

CFS-VR: Continuous Flash Suppression using a Virtual Reality Headset

User Guide and Manual

Copyright © 2024, David S. March

Last updated 07/2024; please see website for most recent version.

Please cite: March, D. S. & Olson, M. A. (2025). CSF-VR: Software for studying unconscious cognition within a VR headset using continuous flash suppression. *Social Cognition*.

A zip file containing CFS-VR and all associated documents can be downloaded from <https://www.marchlab.org/cfs-vr>.

Table of Contents

Introduction and Getting Started	1
Section 1: The Study File, Study Runner GUI, and Running Study GUI.....	2
1.1 The Study File	2
1.2 Study Runner GUI	2
1.3 During the Study.....	4
Section 2: The Mask File and the Mask Maker GUI.....	6
2.1 The Mask File	6
2.2 Mask Maker GUI.....	8
2.3 Save Masks Menu	10
Section 3: The Palette File and the Palette Maker GUI	12
3.1 Palette File	12
3.2 Palette Maker GUI	13
3.3 Upload/Add Palette Menu	14
Section 4: Trials: The Building Blocks of a Study	15
4.1 The Basic Structure of a Study	15
4.2 The Basic Structure of a Trial	15
4.3 Defining the Trial Type.....	18
Trial Types	18
Trial Type 0: Instruction Trial	18
Trial Type 1: Break Trial	19
Trial Type 2: Response Trial.....	19
Trial Type 3: Noise-as-Mask Trial.....	19
Trial Type 4: Object-as-Mask Trial.....	20
Trial Type 5: Multi-Stimulus, Noise-as-Mask Trial.....	20
Trial Type 6: Multi-Stimulus, Object Image-as-Mask Trial.....	20
Trial Response Feature	20
Section 5: Constructing a Study File	22
5.1 Columns in the Study File.....	22
5.2 Randomization: Trials, Blocks, and Conditions	26
Section 6: Static Images, Image Lists, and the Location of Images or Image Lists	28
6.1 Defining the Static Image(s).....	28

Single Items	28
Image Lists.....	28
6.2 Defining the Location of the Static Image(s)	29
Single-Item Trials (Trial Types 1-4).....	29
Multi-Stimulus Trials (Trial Types 5-6)	30
Section 7: The Output File.....	32
Section 8: A Tutorial of the User Template File	33
Section 9: Other Features.....	37
9.1 Running through the Command Line.....	37

Introduction and Getting Started

CFS-VR requires a Windows PC capable of supporting a Meta brand headset via a Quest Link capable cable which connects the PC to the VR headset. Although any Meta brand headset will work, we recommend the Quest 2 or Quest 3. Running a VR headset via Quest Link requires a suitable graphics card (refer to Meta documentation for more details). *Prior to loading CFS-VR, you will need to activate QuestLink in the Oculus program.*

You downloaded a zip file containing CFS-VR and all associated documents. We recommend you unzip and copy the folder to another location. You may also want to create a shortcut icon to place on your desktop. Within the main CFS-VR folder is a subfolder named UserGuide. The UserGuide folder contains the documentation PDF as well as the three comma separated value (CSV) files that you must have and can copy and edit to conduct a study in CFS-VR. Combined, the Study, Mask, and Palette CSVs allow for customization of the trial structure and the noisy image masks. It is not necessary to edit the mask and palette files, as they come pre-loaded with several palettes and mask variations. But it is a good idea to get familiar with the ways in which those features are specified. This section will introduce you to each of the files you need to run a Study.

We will start by introducing you to the graphic user interfaces (GUIs) you will encounter within CFS-VR. Each GUI is associated with one of the three necessary files. Some of the terms and concepts in this section may seem a bit unclear at this point. But do not worry, this section is only to familiarize you with the layout and serve as a reference. Much more information about the functionalities contained within each GUI screen will be presented later on when you are introduced to each of the terms.

Section 1: The Study File, Study Runner GUI, and Running Study GUI

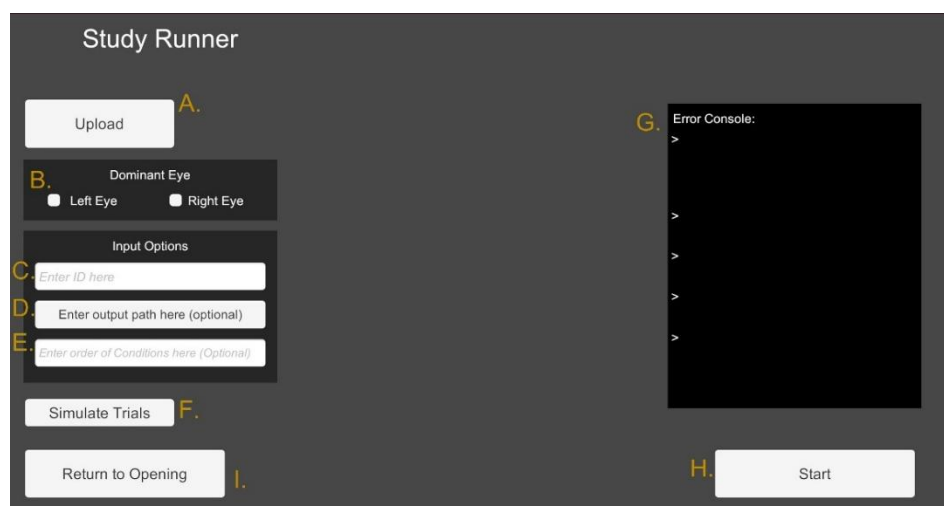
1.1 The Study File

A starter study file was installed when you installed CFS-VR. You can find that file named *UserTemplateCFSVR.csv* located in *UserGuide* folder within the main *CFS-VR* folder. There is also a read-only version of the template file that you can use as a starting point if you at any point need a clean starter study file.

The Study file is where you build your study. You can name the Study file whatever single word you want, but *you must maintain the default column order within the file*. Information on the individual columns themselves is found later in this guide. That being said, you can rename the column headers to whatever you like (but *they cannot contain commas*). The program ignores the first row and only expects that certain information is in certain columns. We suggest that you make a copy of the sample study file and place it in your study folder before editing it to build your study.

1.2 Study Runner GUI

The Study Runner is where you upload your study file, set the participant's eye-dominance, and enter the participant ID. You must select the dominant eye of the participant.¹ This is the eye to which the flashing mask image will be shown. This is also where you enter the order of conditions, if you wish to set one. You can see these various fields in the figure of the Study Runner GUI.



¹ Form a circle with your hand and extend your arm in front of you. Look through the circle at a distant object with both eyes open. Then close one eye and then the other, noting the object's position each time. The object will remain centered when your dominant eye is open. NOTE: some people have weak or no eye-dominance. In such cases, we suggest using the right-eye as the dominant eye.

- A. Before running a study, you must select your study file.
- B. You must select the dominant eye of the participant. Importantly, this is the eye to which the flashing mask image will be shown.
- C. You must give the participant an ID. This ID will be used within the data output file. The ID is also the name of the output file. This can be a string, number, or combination.
- D. You have the option to select an output folder in which to save the output file. If you do not specify a folder, the data output file will be written to the same directory as the uploaded study file.
- E. If your study has multiple within-subject conditions, you can select which conditions are presented and the order in which the conditions are displayed within the study as it runs. These conditions must be defined within the study file in column A. The condition order must be specified as one combined number. For example, if there are 3 conditions defined in the study file, you can input “231” and the conditions will present in that order. If you leave this blank, the conditions will present in the order they are entered within the study file. Or, if you wish to only present one condition, for example condition 1, you can simply enter “1”.
- F. For error checking purposes and to ensure the data is written to the output data file in a satisfactory format, you can simulate the uploaded study. When the simulation is complete, generally very quickly, a message will print to the console. An output file will be produced containing simulated trials. This will be placed in your output folder or the main study folder if you do not specify an output folder. This will also contain an extra column (i.e., not typically contained in log files) where any noted errors will be listed. This file will be located in the output directory you specified in step D and appended with a “_Simulate”.

Importantly, many errors defined within your study will not stop it from running but may be relevant to the participant's experience. Therefore, simulating is a good idea. You can also run through the entire study and explore the output log file. Either way, please check for errors before beginning data collection.

- G. The console will display any critical errors preventing the study from running correctly. These are things like being unable to read the study due to critical

mistakes, not being able to find a mask.csv or color palette, or not being able to find images in the /Stimuli folder.

H. The Start Button simply starts the study.

*****Upon start, there will be a delay as the program builds all of the masks needed for the entire study. There is a progress bar that will let you know when the masks are finished being created and stored.**

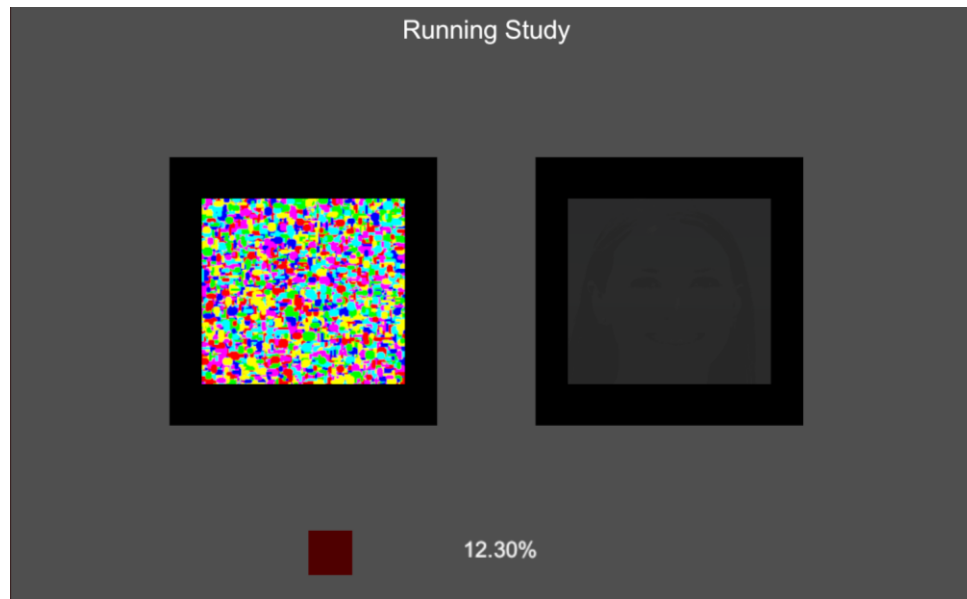
Once the masks are built, the program will pause on a blank, black screen, prior to starting the first study trial. This pause is hard-coded and not something you can turn off. This is built so that you can “start” the study and let it process through building the masks prior to actually putting the headset on the participant’s head. This way, the participant does not have to wait in the dark within the headset while the masks are built.

The experimenter must press the spacebar to advance beyond this pause screen and to begin the study.***

I. Return to opening takes you back to the beginning of the program. From there you can choose to return into the Study Runner or enter the Mask Maker.

1.3 During the Study

Once you start the study, CFS-VR creates all of the masks to be used during the entirety of the study. *This can take some time depending on the number of defined profiles and their specifications. So, there is often a 30 second to 1 min delay after pressing Start but before the study is ready to run.* During the study, you will see a Running Study GUI displaying what the user is seeing through each eye and an approximate indication of their progress in terms of percentage complete.



Important: upon running through the entire study file, the study will restart . We therefore recommend making the last trial of your study a Break Trial (type 1; see section 4.3) with a long duration (e.g., 60000ms) and an image instructing the participant to remove the headset (or whatever you like). The researcher can end the study here (and at any time during the study) by pressing the esc key. All data prior to escaping the study is recorded.

Section 2: The Mask File and the Mask Maker GUI

2.1 The Mask File

An editable starter mask file was installed when you installed CFS-VR. You can find that file named mask.csv located in the UserGuide folder.

IMPORTANT: *The mask file must be named “mask.csv” if custom color palettes are used in noisy masks and it must be in the same directory as the study file.*

The Mask file is where you will define mask profiles for use during the main study. Within the study file, you must refer in Column N to a mask profile for use on each masked trial (e.g., whatever you have named your mask profiles in the mask.csv file). That profile must be contained within the mask file stored in the same directory as your study file. The mask file will reference a palette file that must also be stored in the same directory as your study file. In your mask file, each profile must contain values identifying which palette profile to use, shape, size, and density values.

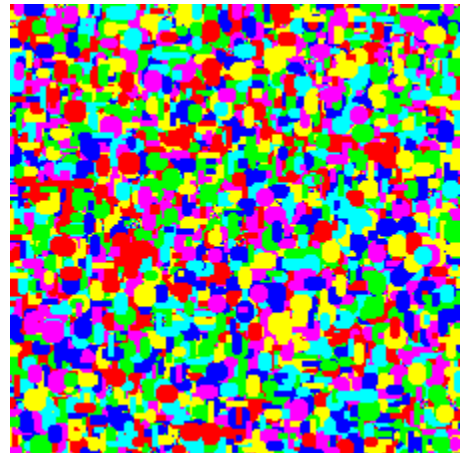
It is recommended that you first use the Mask Maker program to define and test out parameters that define a mask profile. To make that profile, you can use one of the pre-built palettes profiles or make your own in the Mask Maker. You can save profiles to your mask file directly in Mask Maker or take those values and manually input them.

Although you can define mask profiles and palettes manually as described below directly in the mask.csv file and colorPalette.csv, it is often easier to use the Mask Maker program and Palette Maker sub-program to determine the correct parameters and then transfer those into their associated csv (either mask or palette). We provide much of the below information so you understand the contents of the mask and palette files as well as how to manipulate those manually should you wish to do so.

Do note, there is a default, built-in mask that uses a default palette that you can use by inputting a value of 0 in the masks column (N) of your study file. The default mask uses an ellipse shape with a bright, neon color pallet. You can see a default mask sample in the image to the right.

These are the color values used in the default palette:

```
255, 0, 0      //red
0, 255, 0      //green
```



0, 0, 255	//blue
255, 0, 255	//magenta
255, 255, 0	//yellow
0, 255, 255	//cyan

If you want to define your own masks, the next section will define each specification within the mask file.

Although you can define as many mask types as you want in the mask.csv file, only five noisy mask profiles can be used within the study. This is because CFS-VR builds all the masks the study will need once you start the study, and without limiting the number it would take a very long time.

Below is a description of the columns within the mask file:

- A) Name:** This is a name you give to a mask profile. It can be words, numbers, or a combination of both words and numbers. *It cannot contain commas.*
- B) Palette:** You must indicate which palette from the palette file that the profile will use for making the noise mask. You can use the custom palette (0) or can reference a custom palette that you have created and that has been stored in the palette file.
- C) Shape:** This column is used to define what shape will populate the noise mask
 - 1 - Ellipse
 - 2 - Rectangle
 - 3 - Triangle
 - 4 - Pixelated
 - 5 - Circle
 - 6 - Square
 - 7 - Mixed (all shapes equally represented)
- D) Pixelated:** This option allows you to choose between a pixelated background (using specified color palette) or a white background.

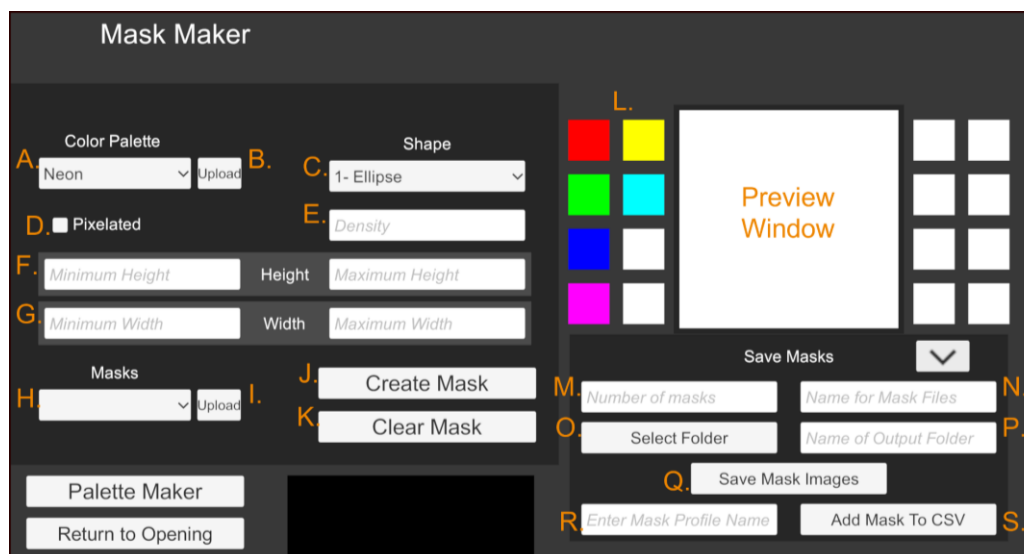
Together, width and height determine the size of each shape within the mask. You must set a minimum and maximum for both width and height. A number within that range will be randomly selected for each shape in the mask.

- E) Minimum Width:** minimum width in pixels of each shape

- F) Maximum Width:** maximum width in pixels of each shape
- G) Minimum Height:** minimum height in pixels of each shape
- H) Maximum Height:** maximum height in pixels of each shape
- I) Density:** This parameter specifies the number of shapes that will be displayed within the bounds of the mask. *Do note that the more shapes that must be drawn, the longer it will take for the study to load.*

2.2 Mask Maker GUI

Within the main CFS-VR program, or via the standalone Mask Maker program, you can access a platform to determine the parameters for different mask profiles. Once you have decided on the parameters for a mask profile, you can save this profile to your mask.csv file or manually enter this information into the mask file. Within the Mask Maker is where you will fill in all of the customizable specifications for use in creating your masks.



- A.** This is a dropdown list for different color palettes. Neon and Black & White are two palettes that are preloaded. When you load a palette file, all of the palette profiles contained therein will appear in this dropdown. When you select a palette, the associated colors will be displayed on the palette boxes (L). *If you want to edit a mask file, you must upload the palette file containing the profiles referenced in the mask file prior to uploading the mask file.*

- B. This is where you upload a color palette file to be used in the Mask Maker. This will also upload the palette into the Palette maker for editing.
- C. This dropdown lists all the different shapes that can be used within the mask.
- D. If you check this box, the background of the mask will be pixelated. This helps to fill in any white space if you do not create a highly dense mask.

The total height and width of the mask is set at 128 x 128 and is scaled up to cover the 256 x 256 suppressed image. These next three fields (Density, Height & Width) interact to determine how many shapes are visible within the boundaries of the mask. Density defines the total number of shapes; not all may be visible if density is set high and they stack on top of one another. Height and Width define the size of each shape presented within the mask. Together they indirectly determine how many shapes are visible within the boundaries of a mask.

- E. Text entry field for the density of shapes in the mask. The density defines how many shapes will be forced to fit within the borders of the mask, even if they are forced to stack on top of one another. For example, if you set density too low, you would see white space. **If you set this number very high, the masks will take very long to build.** So, we recommend starting low and working your way up to your desired density. A good rule of thumb with a height and width typically in the 5-15 range would be a starting density value of around 1000.
- F. Height input fields for the shapes on the mask. This unit is in pixels. You must set a maximum and minimum height. A number within that range will be randomly selected for each shape in the mask. This height value and the width value define an individual shape. We think is a good idea to start around 5-15 and adjust from there. The max is 128, but you should never set the max that high.
- G. Width input fields for the shapes on the mask. This unit is in pixels. You must set a maximum and minimum width. A number within that range will be randomly selected for each shape in the mask. This width value and the height value define an individual shape. We think is a good idea to start around 5-15 and adjust from there. The max is 128, but you should never set the max that high.
- H. If you have uploaded a mask file, here is where you can select a mask from that file. This will autofill all input fields with that mask's specifications as defined in the mask file. Importantly, before uploading the mask file, you must upload the palette file referenced within the mask file.

- I. Here is where you can select a “mask.csv” file to upload into the Mask Maker for editing. Each mask profile defined in that file will appear in the dropdown (H).
- J. This button will create a mask with the current input specifications. You will see a sample of that mask in the preview window.
- K. This button will clear the preview window but not the entered specifications.
- L. The palette boxes display all of the colors associated with the selected palette.

2.3 Save Masks Menu

When you open Mask Maker, this sub menu is hidden. Simply click the down arrow and it will appear. Click the up arrow to make it collapse. This menu is where you can save masks as PNG files to your device to be used in other projects/programs or save your mask profile into your mask file.

- M. If you wish to save masks as individual image files, you must specify the number of output masks you would like saved to your device. Each mask will be a unique image created from the parameter ranges you set. For example, if you set the number of M to 20, Mask Maker will use the input parameter values you have set (palette, density, height, width, etc.) to make 20 unique masks. Remember, you are entering ranges for height and width, of which a random value in that range will be chosen for each mask. The location of each shape on the mask is also random, and the color of that randomly located shape is also random. So, no two masks will be identical, and they will vary along several dimensions.

A count of this number will append to the name you choose (N).

- N. Before saving a mask as a file, you must give that mask a name. An example would be “blueCircleMask”, and each mask file would be named “blueCircleMask” appended with a number counting from 0 to the number specified in (M). For example, if you name the mask blueCircleMask and choose to save 5, they will be saved as blueCircleMask0.png, blueCircleMask1.png, etc.
- O. To set the output folder in which to save your masks, use the explorer window.

P. If you want to save your masks into a new folder (which will become a subfolder of the output folder specified in [O]), you can name that folder here. If left blank the files will be saved to the output location specified in (O).

Q. This button will save the masks to your computer.

If you wish instead to save the newly created mask profile to your mask.csv, use the below buttons:

R. Here is where you can name the just created mask profile to add to the mask file. You can name it anything you like, including numbers and letters. *It cannot contain commas.*

S. If you want to save the currently created mask profile to the mask.csv that you uploaded into the Mask Maker (I), this will write a new mask profile to the end of your uploaded mask file.

Section 3: The Palette File and the Palette Maker GUI

3.1 Palette File

An editable palette file was installed when you installed CFS-VR. You can find that file named colorpalette.csv located in the UserGuide folder.

IMPORTANT: *The palette file must be named “colorPalette.csv” if custom color palettes are used in noisy masks, and it must be in the same directory as the study file.*

The palette file is used for CFS-VR, Mask Maker (GUI enabled noise mask builder), and the Palette Maker program. The palette file is where you will define color palettes for use within each mask profile. Within the palette file, the first column defines the profile, and then each subsequent grouping of three columns (e.g., BCD, EFG, HIJ, KLM, NOP, QRS, TUV, WXY, ...) defines a single color within that profile. Each grouping must contain the red, green, and blue value of that color, in that order. You can list as many colors as you like within your palette, but we recommend limiting the amount to 6. You can enter RGB values pulled from any source, but we recommend using the Palette Maker to define and refine your specifications.

To make the organization of the palette file simple, both CFS-VR and the Mask Maker software ignore rows 1 and 2 of the palette file. As you will see in the palette file that installed with CFS-VR, after the profile name in column A, the top row contains a repeating list of red, green blue, red, green, blue, ... and the second row demarcates color 1 from color 2 etc. In actuality, you can have whatever text you like within the cells of the top two rows as they are only there for organization purposes. But the profile name and RGB values must begin in row 3.

Below is a breakdown of the columns present in the palette file. Remember, after row D, each three rows (e.g., EFG) represents another grouping of RGB values:

- A)** Palette name: This is a name you give to a palette profile. It can be words, numbers, or a combination of both words and numbers. *It cannot contain commas.*
- B)** Red: Enter the red RGB value of the color you want to add
- C)** Green: Enter the Green RGB value of the color you want to add
- D)** Blue: Enter the Blue RGB value of the color you want to add

You can continue to enter as many RGB values as you like to create a color palette. We recommend a minimum of 5 for a sufficiently noisy palette, but you can enter as few as 1 and as many as you like.

Pro tip: if you want one color to appear more than others, you can add it multiple times within the same profile and it will be over-represented in the final product mask. For example, if you want a black and white palette that is mostly black, enter the RGB values for black twice and for white once.

3.2 Palette Maker GUI

You reach the Palette Maker through the Mask Maker, and vice versa, but it makes sense to start with the Palette Maker so the palettes you create are available in Mask Maker. Within the Palette Maker, you can load a palette file to see it and edit it. Or you can create an entirely new palette file for use in your mask files. The main menu is where you will customize the colors that will go into your palette.

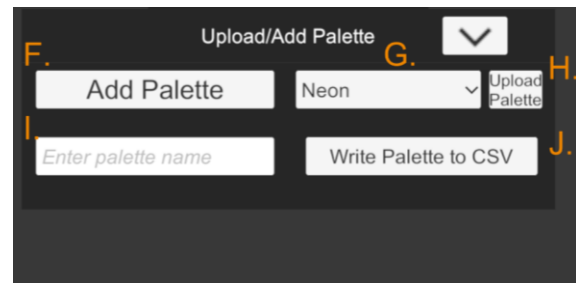


- A. This is a sliding scrollbar that you can use to adjust the red value of the color.
- B. This is a sliding scrollbar that you can use to adjust the green value of the color.
- C. This is a sliding scrollbar that you can use to adjust the blue value of the color.
- D. This will add the color you have created to the next unused palette box on the right or below.

- E. This is the palette box that displays all the colors in your current palette. If you want to change a specific color in the palette, you can click and select it in the palette display. Clicking any color will automatically delete it from the palette and write its RGB values to the sliders on the left. You can add the same color back by pressing “Add Color” or alter the color and add the new color using the same button. This is also how you can remove any unwanted colors from your palette. Simply click on a color and it will delete from the palette box.

3.3 Upload/Add Palette Menu

When you open Palette Maker, this sub menu is hidden. Simply click the down arrow and it will appear. Click the up arrow to make it collapse. This menu is where you will send a palette for accessing through the Mask Maker, save a palette for use within the Palette Maker, and add a palette into an existing Palette file.

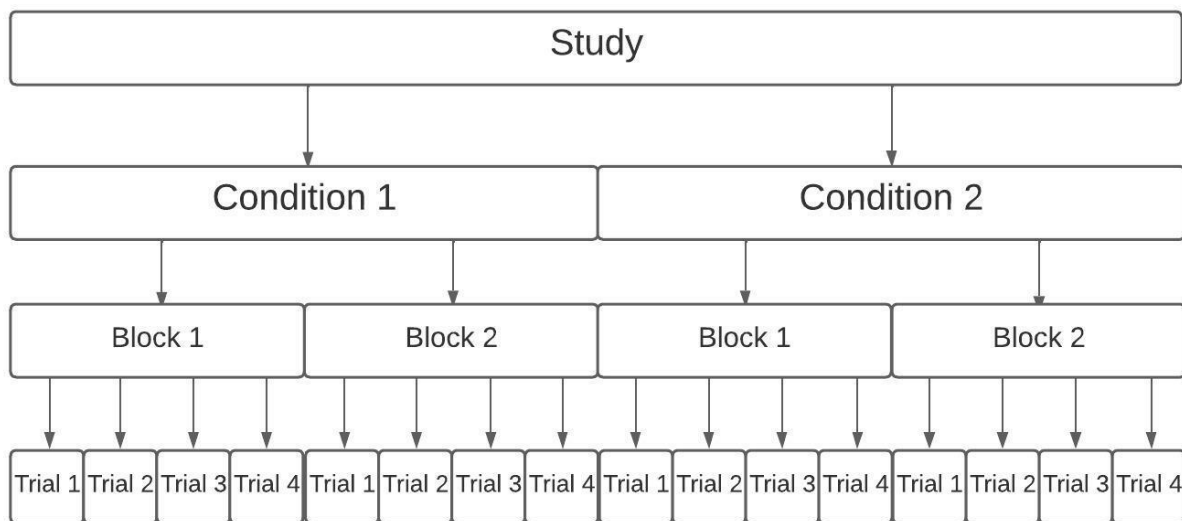


- F. This button has two functions. First, clicking it will add the current palette to the Palette Maker list of available palettes (G) that you are currently working with. You will see your newly defined palette profile listed under the dropdown. And this newly created palette profile will become available in list of palettes active in the Mask Maker program. Importantly, this does not add the palette to your Palette file; to do that you must use J.
- G. This dropdown lists the available palettes you can load into the Palette Maker. Some of these palettes are available by default (neon & black/white). If you have defined other palette profiles, they will be available here. Or, if you have loaded a palette file using H, all palettes in that file will be available in this dropdown. This is how you choose a preexisting palette to which you would like to make changes. Importantly, this will not edit the palette in the palette file, but instead is used to make changes to the working palette (which will be overwritten).
- H. This button allows you to choose a “colorPalette.csv” to upload to Palette Maker. This will add every palette profile in that file into the list of editable palettes (G).
- I. This is where you can name the palette profile before adding it to the Palette Maker or palette file. *It cannot contain commas.*
- J. This button adds the palette you have created to the currently loaded palette file.

Section 4: Trials: The Building Blocks of a Study

4.1 The Basic Structure of a Study

Now that you know the basic elements of the software and various CSV files you will use, it is important to understand the basic structure of the Study. The design of your study be conceptualized as several hierarchically organized smaller subunits. As illustrated in the figure below, the highest level is the Condition, within which there are Blocks, within which there are Trials. This hierarchical organization allows for several types of randomizations to occur. Importantly, as discussed below in Section 5.2, *randomization is contained within each unit*. The smallest unit is the trial, which is basic element seen by your participants. The next sections detail the basic elements of the trial as well as each of the parameters that together determine the structure and dynamic of a trial.



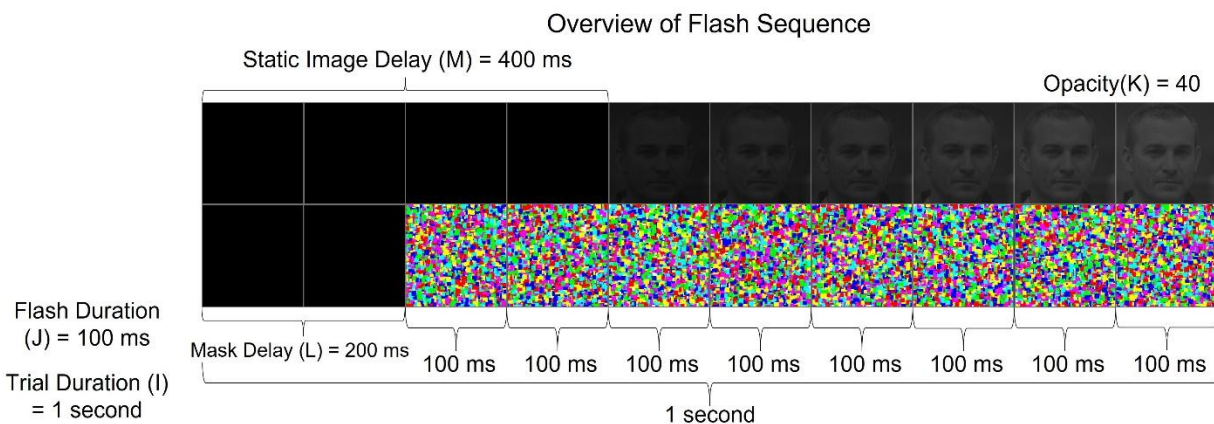
4.2 The Basic Structure of a Trial

The basic elements of a trial can be broken down into (a) what is presented to the dominant eye, (b) what is presented to the non-dominant eye (i.e., what is suppressed), and (c) what are the time parameters associated with each element of (a) and (b). CFS-VR allows you to specify many combinations of trial structure and timing. In your Study file, each trial is defined by the specifications you set. You will get much more information on those options in subsequent sections. For now, it should suffice to simply be aware of the basic elements that make up a trial.

The entirety of a single trial is called the Trial Duration (i.e., the length of a single trial). The Trial Duration contains several cycles of Flash Durations (i.e., the length of a single flash). Flash Durations can also contain a Blank Period (i.e., the length of black screen ending each Flash Duration prior to the onset of the next flash). Each of these are definable parameters set by you the user. You are likely thinking, “what the hell?”, but bear with us. The below description and diagram will help make this clearer.

It is helpful to think of the trial itself as divided into multiple cycles. The number of cycles is determined by dividing the Trial Duration by the Flash Duration. **Therefore, it is crucial that the Flash Duration is a divisor of the Trial Duration.** For example, if you set the Trial Duration to 1000ms (1 second), the Flash Duration must be a divisor of 1000. Indeed, you can set the Flash Duration to 100 ms, 200ms, 250ms, or any number equally divisible into 1000. In our experience, 100ms is a good starting point for determining whether your design results in suppression. If you set it too low, suppression will suffer. (Theoretically, the lowest you can set this value is limited only by the refresh rate of your headset. But setting it too low diminishes CFS.)

In Example 1, the Flash Duration is set to 100, which means there will be 10 cycles within each 1000ms Trial Duration.

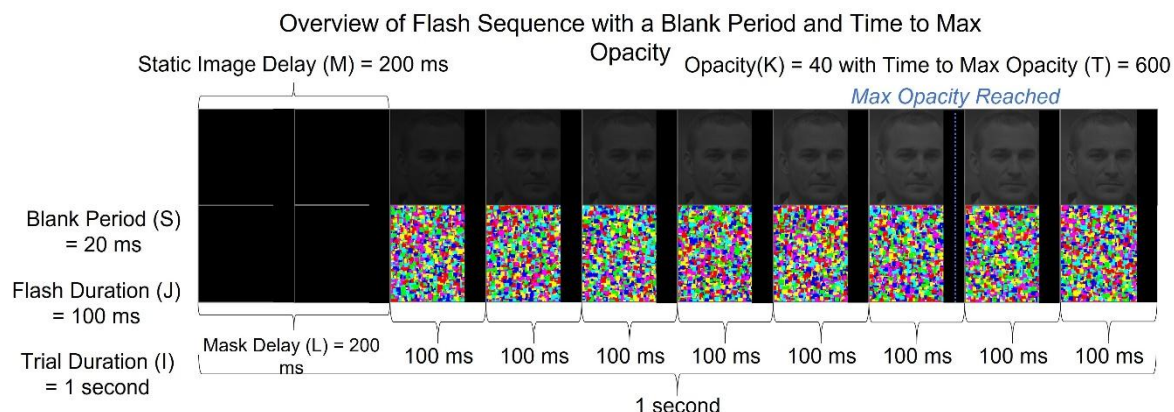


Example 1.

In Example 1, you can see a 1-second trial (1-second Trial Duration set by the user), which contains 10, 100-ms Flash Durations. The Flash Duration describes the length of time a mask image (either noisy or object) is presented to the dominant eye before flashing. It also describes the length of time between increases in opacity of the image(s) presented to the non-dominant eye. In this example, max opacity is set to 40%, which will linearly increase from 0 at the onset of the suppressed image to the end of the trial.

The *Mask Delay* indicates when the mask begins to appear. The *Static Image Delay* indicates when the suppressed object image begins to appear. As you can see, these do not need to be set to the same value. Both the Static Image Delay and Mask Delay values must be in a multiple of the Flash Duration to ensure each will onset with the onset of a new cycle. In this example, there is also a user-specified Mask Delay of 200ms, which is the time before the mask appears (after two 100ms Flash Duration cycles). This means that the first mask Flash will begin to appear on the 3rd cycle. There is also a Static Image Delay of 400 ms, which is the time before the suppressed image appears (after four 100ms Flash Duration cycles). This means that the first instantiation of the suppressed image will begin to appear on the 5th cycle.

As Example 1, if you want the flash to begin appearing prior to the onset of the static image, you can set the Mask Delay to a number less than the Static Image Delay (though again, both must be in multiples of the Flash Duration; for example, setting the Mask delay to 200 and the Static Image Delay to 400 will ensure 2 cycles between the onset of the mask and the onset of the static image). Or, as in Example 2, you can set these values the same and they will onset on the same cycle.



Example 2.

In Example 2, you can again see a 1000 millisecond trial, which contains 10, 100-ms Flash Durations. In this example, there is also a Blank Period, which here is set to 20 ms. The Blank Period is the “break” between flashes. The optional Blank Period specifies the duration in milliseconds from the end of the Flash Duration when the mask and suppressed image will be off the screen (i.e., for how long the screen will be blank). Essentially, it is the time at the end of a Flash Duration that both dominant and suppressed images are cleared from view. This short burst of blackness can add to the flash dynamic. If you want a more intense “flash” between cycles, you should set this value. If you do not, the masks will flash back-to-back.

Here there is a Mask Delay of 200ms, which is the time before the mask appears, and now the Static Image Delay has been set to 200ms. By setting the Static Image Delay to the same value as the Mask Delay, they will onset at the same time during the trial. This example has a max opacity set of 40, but there is also a Time to Max Opacity set of 600. By adding this parameter, opacity will linearly increase from 0 at the onset of the suppressed image (which occurs 200ms into the trial) to reach the max opacity set at 800ms.

4.3 Defining the Trial Type

Now that you (hopefully) have a good idea of the terminology and the basic options available within CFS-VR, you can determine what type of trial you would like to present. Trial types can change between trials, between blocks, or between conditions. You can mix and match in any way you like. The various trial types should allow you maximum flexibility to design your study in any way you want. Some trials present only images with no flashing. Others present options for participant response. Some present the noise-mask to the dominant eye, whereas others present a static object to the dominant eye. And whereas most trial types present one image to the non-dominant eye, there are other trial-types that allow you to present two stimuli as suppressed images. By using the other parameter options available, you can create many different dynamics within each of these trial types.

Trial Types

Trial Type 0: Instruction Trial

During these trials, the same static image is displayed to both eyes to enable participants to read the content. Although participant input is necessary to proceed, the image will remain on the screen for a minimum of the duration specified. The participant can continue the trial by pressing the right trigger on the controller or the spacebar.

These trials require you to define the instruction image and the specified *minimum* duration, as shown in the figure below. Here you can see 1 instruction trial image that will last for 1000ms. The trial will move on once the participant presses the right trigger or spacebar. Although the block this instruction is in was randomly ordered, this trial will always appear in its current location as trial_randomization is set to 0.

A	B	C	D	E	F	G	H	I
condition	condition	block	block_rand	trial_type	trial_input	trial_rand	static_image	duration
1	0	1	1	0	1	0	NoiseAsMaskIntro.png	1000

Trial Type 1: Break Trial

Break trials *do not* require any input from the participant to proceed. The trial will end at the duration specified. These trials require you to define the break image and the specified duration, as shown in the figure below. Here you can see 1 break image that will last for 2 seconds. After that time, the study will continue to the next trial.

A	B	C	D	E	F	G	H	I
condition	condition	block	block_rand	trial_type	trial_input	trial_rand	static_image	duration
1	0	1	1	1	11	0	break.png	2000

Trial Type 2: Response Trial

In response trials, participants are required to input a response to proceed to the next trial. The response options correspond to up, down, left, and right on the D-pad on a controller or the arrow keys on a keyboard. Responses will be printed as up, down, left, and right in the output unless other values are specified in columns O-R as seen below.

Note: any type of trial in the program can be converted into a response trial. See the section on the Trial Response Feature for more information.

E	F	G	H	I	J	K	L	M	N	O	P	Q	R
trial_type	trial_input	trial_rand	static_image	duration	flash_dur	opacity	mask_del	static_del	mask	up	down	left	right
2	10	0	response1.png							Good	Bad	Neutral	Uneasy

Trial Type 3: Noise-as-Mask Trial

Noise mask trials present a static image to the non-dominant eye which is suppressed by several noisy mask images presented to the dominant eye. The combination of these two elements results in continuous flash suppression (CFS; i.e., the suppression of the static image by the noisy image). These trials utilize noisy masks generated at the beginning of the study by the built-in Mask Maker program. These masks are defined by you in the mask file. There you can customize the types of masks used in the study. See information on the Mask Maker GUI and the mask file for more information on customization. If you set the mask column (N) to 0, the default mask will be presented. The format of these trials is shown below.

E	F	G	H	I	J	K	L	M	N
trial_type	trial_input	trial_rand	static_image	duration	flash_duration	opacity	mask_delay	static_del	mask
3	2	1	\$contrastPeople.txt	1000	100	50	200	400	AutumnLeaves

Trial Type 4: Object-as-Mask Trial

Object mask trials present an object image to the dominant eye to function as the mask (instead of a noisy image, as in trial type 3). The format of these trials is shown below.

E	F	G	H	I	J	K	L	M	N
trial_type	trial_input	trial_rand	static_image	duration	flash_duration	opacity	mask_delay	static_delay	mask
4	2	1	&contrastPeople.txt	1000	100	50	200	400	#nonColorGraded.txt

Trial Type 5: Multi-Stimulus, Noise-as-Mask Trial

Multi-stimulus noise-as-mask trials present two object images to the suppressed eye. You must specify in column (H) the two images or image (txt) files to pull images from. These file names or image lists should be entered in column (H) separated by an underscore (_). By default, the two object images are displayed next to one another as shown in the figure below.

These paired object images are masked by noisy images generated by Mask Maker using any parameters you have set. If you set the mask column the 0, the default mask will be presented. Or, you can use custom masks by entering a specific mask profile.

E	F	G	H	I	J	K	L	M	N
trial_type	trial_input	trial_rand	static_image	duration	flash_duration	opacity	mask_delay	static_delay	mask
5	6	1	\$contrastThings.txt_&contrastPeople.txt	1000	100	50	200	400	0

Trial Type 6: Multi-Stimulus, Object Image-as-Mask Trial

Multi-stimulus image trials present two defined object images as the suppressed image. Like trial type 5, these object images are displayed adjacent to one another. Distinct from trial type 5, trial type 6 employs an object image as the mask (N) instead of a noise image. Randomization for object image masks does not have to be the same as the randomization for suppressed static images. The format of these trials is shown below.

E	F	G	H	I	J	K	L	M	N
trial_type	trial_input	trial_rand	static_image	duration	flash_duration	opacity	mask_delay	static_delay	mask
6	2	1	&contrastPeople.txt_#contrastThings.txt	1000	100	50	200	400	&nonColorGraded.txt

Trial Response Feature

Along with the native response trial type 2, trial types 3-6 can be converted into a response trial by using the response columns (O-R) in the study file. Participants can use the direction-pad on any controller or arrow keys on a keyboard to provide input at any point during the trial.

By default, a response trial will end once a response is entered unless column 'V' is used to indicate a multi-response trial. Entering a value of 1 in column 'V' indicates that

the trial should continue even after input is provided. If the participant enters more than one response, all responses and their response times are recorded in the output file, separated by an underscore. See the figure below for an example input file.

N	O	P	Q	R	S	T	U	V
mask	up	down	left	right	blank_per	time_to_r	quadrant	multi_response
	0 Good	Bad	Neutral	Uneasy	50			1

Section 5: Construction of a Study File

The entire study is built within a CSV file. A demonstration study file called UserTemplateEditableCFSVR is available in the UserGuide folder. It is a good idea to start from this file when building your studies. There are several columns that allow you to tailor the study in several ways. The first set of columns (A-G) describe the structure of the study in terms of conditions, blocks, and trials, as well as whether there are any within block (i.e., trial-level) or between block randomization. Columns H-N describe the trial structure by defining the dynamics of the suppressed image, the mask, the length of the trial, and the dynamics of the mask. The remaining columns define features specific to a subset of trial types. You can find details on every column described below.

Important Note: *The title of the columns is for the benefit of the user; they are not read by the software. Only the location (i.e., the specific letter) of each column is considered by the software. For example, you can rename Column A from Condition to Group, and the running of your study will not be affected. But it will expect Condition-type information to be contained in column A. The same is true for all columns. You can rename them whatever you want as it is only the type of information contained within each column that is expected by CFS-VR. Importantly, names cannot contain commas.*

5.1 Columns in the Study File

<u>Column</u>	<u>Column Description</u>
---------------	---------------------------

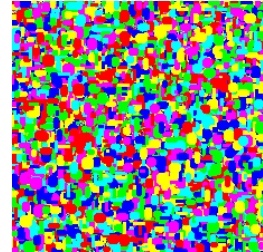
- | | |
|-----------|--|
| A. | Condition: This number defines the condition, the top-level organizational structure that contains a block or blocks which each contain trials. <i>Participants can experience only one condition or multiple conditions depending on what you specify in field E of the study runner screen.</i> |
|-----------|--|

Note: conditions, blocks, and trials must be defined in a numerically sequential order starting at 1

- | | |
|-----------|--|
| B. | Condition Random: Determines whether the order of this condition and other conditions should be randomized. Enter a 1 to indicate randomization, or 0 to indicate not to randomize. |
| C. | Block: Defines the Block which contains one or several trials. |

- D. **Block Random:** Determines whether this Block should be randomized with the other blocks in the same condition.
- E. **Trial Type:** Defines the dynamics of the trial. 0-6 are acceptable inputs.
- F. **Trial:** Numbering trials in each block. This value is expressed in the output reflecting values indicated here. This is distinct from the Order variable in the output file which displays actual presentation order. These two values may differ due to randomization.
- G. **Trial Random:** If the presentation order of this trial should be randomized with the other trials *in the same Block*. Groups of trials to be randomized will be indicated by using the same value. Multiple values can be specified to create multiple groups of randomized trials within a single block.
- H. **Static Image (i.e., the stimulus) or Image List:** This is the image presented to the non-dominant eye. Defines the actual name of the *suppressed* image file or a text file containing a list of images. If using an image list, several options for pulling images from the list are available and one must be selected.
- I. **Trial Duration:** The length of time that the trial will run in milliseconds. This is the total trial length, including all flashes.
- J. **Flash Duration:** The interval of time within a trial in which a mask will be displayed before increasing opacity/changing masks.
 - The Flash Duration must be an even number divisor of the Trial Duration. For example, if you set the Trial Duration to 1000ms, you can set the Flash Duration to 100 ms, 200ms, 250ms, or any number equally divisible into 1000.
 - The Flash Duration and trial duration of the trial together define the frequency of which the mask is flashed.
- K. **Opacity:** The maximum percent opacity the Stimulus should reach either by the end of the trial, or if column V (i.e., Time to Max Opacity) is used, by that time within a trial. Setting too high a value will inhibit CFS.
- L. **Mask Delay:** How long in milliseconds from the beginning of a trial until the mask will begin to flash. This value must be a multiple of the Flash Duration or 0.

- M. Static Image Delay:** How long in milliseconds from the beginning of a trial until the suppressed image begins to appear. This value must be a multiple of the Flash Duration. The Static Image Delay must be equal to or greater than the Mask Delay (column L) and have a minimum of 1x the flash duration. This ensures that the static image does appear prior to the onset of the CFS mask.
- N. Mask:** Defines which mask will be presented within a trial. The mask varies depending on trial type. For noise mask trials, a value of 0 will render the default colorful mask using the default pallet. (Additionally, if you do not specify a mask in the study file, the program will present the default mask.) A sample of the default (ellipse) mask can be seen in the figure to the right. The mask “name” is defined in the mask file. The name can be a word or a number.



Parameters O-Y are optional and dependent on trial type.

<u>Column</u>	<u>Column Description</u>
O.	Up: In a response trial, this is what will be printed to the output if up is selected. If left blank, the default language “up” will be printed to the output. If you enter any other label, that label will be printed instead.
P.	Down: In a response trial, this is what will be printed to the output if down is selected. If left blank, the default language “down” will be printed to the output. If you enter any other label, that label will be printed instead.
Q.	Left: In a response trial, this is what will be printed to the output if left is selected. If left blank, the default language “left” will be printed to the output. If you enter any other label, that label will be printed instead.
R.	Right: In a response trial, this is what will be printed to the output if right is selected. If left blank, the default language “right” will be printed to the output. If you enter any other label, that label will be printed instead.

S. Blank Period: The time that the noisy image and suppressed static image are *off* the screen *ending* a flash duration. Any trial type can include a *Blank Period* simply by adding a value to column S. This parameter can be any value less than the *Flash Duration*. Refer back to Example Figure 2 for a diagram of a trial including a Flash Duration.

T. Time to Max Opacity: This is the time from the onset of the static image that the max opacity (set in column K) is reached. If this value is not defined, opacity will increase linearly from onset to the end of the trial. If this value is defined, opacity will increase linearly from onset to this defined time. If this value is set to 0, the static image will appear at max opacity on the first presentation.

This value must be a multiple of the *Flash Duration* and greater than the *Static Image Delay* (i.e., come later than the onset of the static image). For example, if you have a 1000ms trial and set the Flash duration to 100, the value in column T must be a multiple of 100. And imagine you set the Static Image Delay to 400ms, meaning the static image will appear on the 4th flash; you must set the value in column T to equal or greater than 400 but less than 1000 so that the image does not reach max opacity prior its onset and prior to the end of the trial.

U. Location: By default, the suppressed image is presented at the maximum size of 256x256 pixels, the same size as the mask image. If this column (U) is left blank, the suppressed image will be presented using this default setting. But you can also present the suppressed image in a specific or random location within the 256 x 256 border. You have several options for doing so which depend on whether you are presenting a single or multi-stimulus trial. Using this setting will result in a smaller form of the image presented.

If you want to define a location, it is important to first determine whether you are using a single stimulus or multi-stimulus trial. The way CFS-VR implements the defined location will be different for each trial type. For example, you can specify a specific space from a preset number of options, or random location can be chosen from those options. This location can change between but also within trials.

V. Multi-Response: Defines whether the trial should end after taking a single response or allowing for multiple responses. Only used if the trial uses the

trial response feature. If you allow for multiple responses, the log file will record all responses and their reaction times separated by a comma.

- 1 if you want to allow multiple responses.
- 0 or blank (the default) if you want the trial to end after a single response.

W, X, Y. Passthrough Columns: There may be situations in which you want to pass through trial-wise information from the study file to the output file for use later on during data analysis. For example, the condition category, trial category, or any other useful information. If you want to passthrough that information, you can simply add it to any or all rows starting with column W and continuing to column Y. You can add a header to each column and give it whatever name you like (without commas). This exact information yoked to each trial will be copied directly to the output file.

5.2 Randomization: Trials, Blocks, and Conditions

As detailed earlier, your study file can be conceptualized as a series of hierarchical units. This hierarchical organization allows for several types of randomizations to occur. Importantly, *randomization occurs within each unit*.

Specifically:

1. Conditions are randomized with other conditions *within the overall study*.
2. Blocks are randomized with other blocks *within the same condition*.
3. Trials are randomized with other trials *within the same block*. For example, if you use a trial_randomization value of 1 for separate groupings of trials contained within two separate blocks, those trials would not be intermixed, but instead would be randomized *within* each of their respective blocks. Yet, we suggest that to keep your groupings more clearly delineated, it is better practice to use distinct values for each group of randomizations, even if they are in separate blocks.

You can use any randomization number and each trial will randomize within their distinct set numbers. So, trials in a group of trial_randomization =1 will not randomize with a group of trial_randomization =2. But the 1's within a block will randomize with the 1's and the 2's within a block will randomize with the 2's, and so on. For organization, you can enter a value of 0 to indicate no randomization (or leave it blank).

All trials within blocks, blocks within conditions, and conditions within the study can be randomized within each level of the hierarchical structure.

However, as shown in the input file example in the figure below, this need not be the case, and you may want some trials to not be randomized. If so, non-randomized trials (those given a 0 value) will remain in the position they are defined. This feature is useful any time you want a trial to remain in its input trial position, such as an instruction trial that should be the first thing people see within a block. It is often useful to anchor instructional, break, and response trials within the larger block structure.

E	F	G	H
trial_type	trial_input	trial_randomization	static_image
0	1	0	NoiseAsMaskIntro.png
3	2	1	#contrastPeople.txt
3	3	1	#contrastPeople.txt
3	4	1	#contrastPeople.txt
3	5	1	#contrastPeople.txt
3	6	1	#contrastPeople.txt
3	7	1	#contrastPeople.txt
3	8	1	#contrastPeople.txt
3	9	1	#contrastPeople.txt
2	10	0	response1.png
1	11	0	break.png

Section 6: Static Images, Image Lists, and the Location of Images or Image Lists

IMPORTANT: All image files must be in a folder named “Stimuli” in the same directory as your study CSV file. Images can only be in JPG/JPEG or PNG format.

6.1 Defining the Static Image(s)

You must provide individual image files for use as static images presented to the non-dominant eye. Technically the image can be any shape, but it will center-cropped to fit into a square shape exactly 256x256 pixels. All stimuli must be contained within a folder named “Stimuli” which must be in the same folder as the main Study. There are two ways in which images are defined within the study, either Single Item or Image Lists.

Single Items

You can specify a single image file in column (H) within the study file and it will be presented on that trial.

Image Lists

Images can be defined within a text (.txt) file placed in the Stimuli folder. This text file contains a list of the image files you want to use in any particular set of trials. Each image file name should be separated by a hard-return with no spaces at the end of the file name. These images must also be in the Stimuli subfolder in the same main folder as the study file or in their own subfolder within the Stimuli folder. If placed in a subfolder, you must indicate the subfolder location within the Stimulus List (i.e., by specifying foldername/imagename.png). When using Image Lists, only one type of randomization can be used within each block.

The input in column (H) must contain the name of the txt file along with a leading symbol defining how to pull from the image list (i.e., in order, or via some type of randomization; see below). Any number of trials can use the same image list; if the number of trials is greater than the number of image file names in the txt file, the program will randomly sample from the list according to parameters below. You can define several image lists containing completely different or overlapping image files.

You must use one of three symbols to specify the way in which the study will draw images from the Image List file. You can specify that it pull images either in the order

they are listed or via one of two types of randomizations specified below. For the sake of these examples, the Image List is named “imagelist”:

- # to pull from the image list in the order images are listed (even if the trials themselves are randomized). For example, #imagelist.
- \$ to pull from the image list randomly *without* replacement. If the number of trials is greater than the number of image file names in the txt file, the list will repopulate once all images have been used. For example, \$imagelist.
- & to pull from the image list randomly *with* replacement. For example, &imagelist.

Other images can be specified using the first method above (i.e., defining a single image file) even if within the block you also use Image Lists.

Also recall from [Trial Type 5](#), you can include multiple stimuli, multiple Image Lists, and even combine randomization types for those Image Lists.

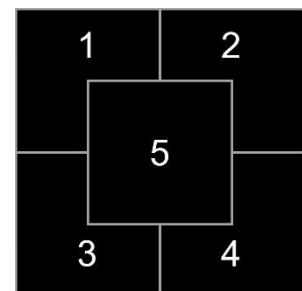
6.2 Defining the Location of the Static Image(s)

By default, the suppressed image is presented at the maximum size of 256x256 pixels, the same size as the mask image. If this column (U) is left blank, the suppressed image will be presented using this default setting. But, if you want to present the suppressed image in a specific location, you have several options for doing so which depend on whether you are presenting a single or multi-stimulus trial.

If wanting to define a location, it is important to first determine whether you are using a single stimulus or multi-stimulus trial. The way CFS-VR implements the defined location will be different for each trial type. Please see below for detailed information.

Single-Item Trials (Trial Types 1-4)

If you are presenting a single-stimulus trial (types 1-4), there are 5 location specific options that correspond to specific areas within the mask, including a centered display. Specifying location 1-5 will present the suppressed single image at 25% of its original size in the defined location. Please see the figure above which displays the five location specific options.



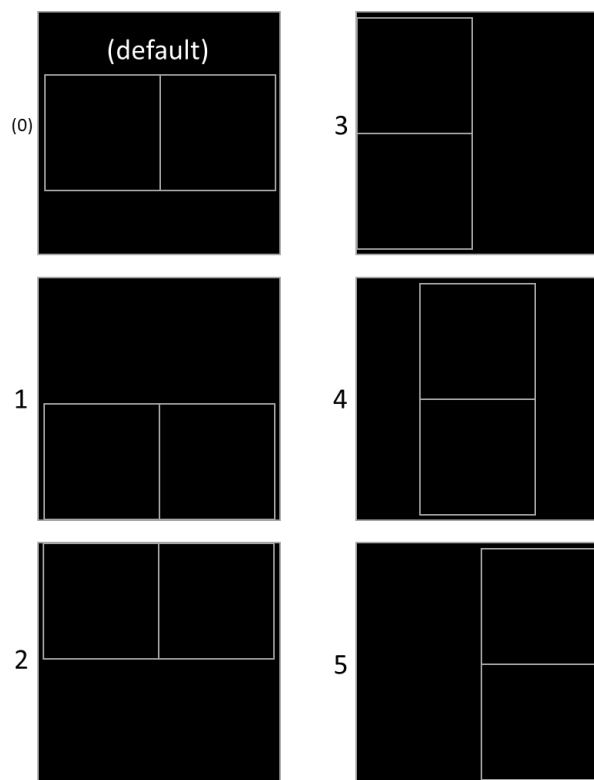
Options 6-9 randomize the location of the suppressed image:

6. Randomly present the suppressed image in one of locations 1-5.
7. Randomly present the suppressed image in one of locations 1-5, and randomly change that location with every new Flash Duration.
8. Randomly present the suppressed image anywhere within the border of the mask.
9. Randomly present the suppressed image anywhere within the border of the mask, and randomly change that location with every new Flash Duration.

Multi-Stimulus Trials (Trial Types 5-6)

There are 6 **location specific** options that correspond to specific areas within the mask. The default multi-stim is option 0 and does not need to be specified. Specifying **location 1-5** will present the suppressed multi-stims at 20% of their original size in the defined locations. Please see the figure on this page which displays the six location specific options.

Importantly, the left/right and top/bottom order is yoked to the order the stimuli were entered in the study file. The first stimulus entered within the study relevant cell will appear as the left/top stimulus of the combinations seen above; the second stimulus entered will appear as the right/bottom stimulus of the combinations seen in the figure on this page.



Options 6-9 randomize the location using different parameters:

6. Randomly present the suppressed images in one of locations 0-5.
7. Randomly present the suppressed images in one of locations 0-5, and randomly change those locations with every new Flash Duration.

8. Randomly present the suppressed images anywhere within the border of the mask and ensure the images do not overlap. The left/right and top/bottom order is not yoked to the order they were entered in the study file.

9. Randomly present the suppressed images anywhere within the border of the mask, and ensure the images do not overlap, and randomly change those locations with every new Flash Duration. The left/right and top/bottom order is not yoked to the order they were entered in the study file.

Section 7: The Output File

During the study, an output file is created and written as the study is conducted. Even if the study ends prematurely via the esc key, all trials to that point are written to the output file. You have the option to select an output folder in which to save the output file. If you do not specify a folder, the data output file will be written to the same directory as the study file. The output file is given the same name as the participant ID.

Trial Count: Indicates the actual order in which trials were presented, regardless of their order in the study file. Randomization of the trials will result in the Trial Count being unique from the Trial Input, which is the order they are listed within your study file.

Trial Type: In your input file, Trial Type is numerical. In the output file, the numerical value is translated into its trial-type text. These are:

- 0 instruction
- 1 break
- 2 response
- 3 noise_as_mask
- 4 object_as_mask
- 5 multi_stim_noise_as_mask
- 6 multi_stim_object_as_mask

CondRand: Will indicate TRUE if condition is randomized. Otherwise, will be FALSE.

BlockRand: Will indicate TRUE if block is randomized. Otherwise, will be FALSE.

Response Time: On response trials, this will indicate the response time. If a single response trial, there will only be one value. If a multi-response trial, there will be multiple values separated by an underscore.

Answer: On response trials, this will indicate the participant's response. If a single response trial, there will only be one value. If a multi-response trial, there will be multiple responses separated by an underscore.

Time to reach max Opacity: Indicates the time you specified to reach max opacity. This value be -1 if nothing is entered.

Multi Response: Indicated TRUE if multi-response was set to 1, or FALSE if set to 0.

Section 8: A Tutorial of the User Template File

When you unzipped CFS-VR, several folders were placed in the CFS-VR main folder. One of those folders is called “UserGuide”, which contains this Documentation, a starter *colorPallete* file, a starter *mask* file, and a demonstration study file called *UserTemplateEditableCFSVR*.

The demonstration study file is meant to function as a starting point for building studies. Within it, you will find many examples of the types of trials, block structure, and randomization features of CFS-VR. These are roughly delimited by Block, though we also mix trial types within blocks to show you the flexible functionality of the study file. We recommend you duplicate this file with a new name as a starting point for building your own studies or simply copy out certain blocks corresponding to the trial types you aim to use in your own study.

To make that process easier, this section will detail each of the features exemplified within the *UserTemplateCFSVR* file. In the study designed within the *UserTemplateCFSVR* file, you will see four conditions, labeled 1-4 in column (A). We discuss each of these blocks separately.

Condition 1

Condition 1 contains three blocks, labeled 1-3 in column (C). You can see in column (D) that `block_randomization` for all three blocks are set to 1. This means that the order in which each of block 1-3 occurs within the time course of the study will be random. Remember, randomization occurs hierarchically and is constrained within the Condition. That means that even though `block_randomization` is also set to 1 for the three blocks in Condition 2, those blocks will not be randomized with the blocks from Condition 1.

Block 1 contains 11 trials. Trial 1 is an instruction trial of [type 0](#), the same static image is displayed to both eyes. The image will remain on the screen for a minimum of the duration specified in column (I). The participant can continue the trial by pressing the right trigger on the controller or the spacebar. Although the block this instruction is in was randomly ordered, this trial will always appear in its current location as `trial_randomization` and is set to 0.

Trials 2-9 are CFS trials of [type 3](#), meaning they are noise-as-mask trials. The stimuli will be pulled from an Image List, as can be seen in column (H). That list is prefixed with a #, meaning the images will be pulled from the list in the order the images are listed. Each of trials 2-9 has a `trial_randomization` (G) of 1, meaning that the order of each trial

will be randomly determined. Remember, just as randomization occurs hierarchically and is constrained within the Condition, it is also constrained within the Block. That means that even though the trials in Block 2 also have trial_randomization values of 1, they will not be mixed the trials from Block 1. These trials will last for 1000ms (I) and have a flash every 100ms (J). The masks that will be shown have a profile name of “Mixed”, which is found in the mask.csv file. Those masks will start to flash on the 2nd cycle, 200ms after the start of the trial. The static image (H) will begin being shown on the 4th cycle, 400ms after the start of the trial. The static image (H) will linearly increase in opacity until 50% opacity is reached coinciding with the end of the trial (because no Time to Max Opacity [T] is set).

Trial 10 is a response trial of [type 2](#). On this trial an image is presented, and participants are required to input a response to proceed to the next trial. The response options in columns (O-R) correspond to up, down, left, and right on the D-pad on a controller or the arrow keys on a keyboard. Your response question must make clear which response corresponds to the response options you set. As you can see in the example to the right, here up = Good, left = Neutral, down = Bad, and right = Uneasy. You can set the values in columns (O-R) to any text and the chosen of those values will be printed in the output data file.

How do you feel about
this person?



	Good	
Neutral	+	Uneasy
	Bad	

Trial 11 is a break of trial [type 1](#). Break trials *do not* require any input from the participant to proceed. The trial will end at the duration specified. These trials require you to define the break image and the specified duration. Here you can see 1 break image that will last for 2000ms . After that time, the study will continue to the next trial, which in this case is the beginning of another block (or Condition, depending on how you have designed your study).

Block 2 is the same structure as Block 1, but now showing the use of the Blank Period (S). These trials still have a Flash Duration (J) of 100ms, but 50ms of that will be blank. The default mask 0 is also used in column (N).

Block 3 is the same structure as Block 2, but now using CFS trial [type 4](#), an object-as-mask trial. Object mask trials present an object image to the dominant eye to function as the mask (instead of a noisy image, as in trial type 3). Here you continue to specify a suppressed image (or Image List) in column (H) and also indicate an image (or Image List) in the Mask (N) column. Here you can see it will be pulling from an Image List

called nonColorGraded and using the # to indicate to pull from the image list in the order images are listed (even though the order of trials are randomized).

Condition 2

Condition 2 contains three blocks, labeled 1-3 in column (C). You can see in column (D) that block_randomization for all three blocks are set to 1. This means that the order in which each of block 1-3 occurs within the time course of the study will be random.

Block 1 contains an instruction trial, followed by several [type 5](#), Multi-Stimulus, Noise-as-Mask Trials. Multi-stimulus noise-as-mask trials present two object images to the suppressed eye. Here you can see specified in column (H) the two images or image (txt) files to pull images from. These file names or image lists should be entered in column (H) separated by an underscore (_). By default, the two object images are displayed next to one another (Condition 3 introduces the Location column where you can customize multi-stimulus locations). These paired object images are masked by noisy images generated by Mask Maker using any parameters you have set.

Block 2 is the same structure as Block 1, but now showing the use of the Blank Period (S). These trials still have a Flash Duration (J) of 100ms, but 50ms of that will be blank. And, in addition to default mask 0 being used in column (N), some trials use masks with a profile name “Mixed”. This shows how you can use multiple mask profiles within the same block simply by specifying them on separate trials.

Block 3 is the same structure as Block 2, but now using CFS trial [type 6](#), Multi-Stimulus, Object Image-as-Mask Trials. Multi-stimulus object image trials present two defined object images as the suppressed image. Like trial type 5, these object images are displayed adjacent to one another. Distinct from trial type 5, trial type 6 employs an object image as the mask instead of a noise image. Randomization for object image masks does not have to be the same as the randomization for suppressed static images. Here you can see it will be pulling from an Image List called nonColorGraded and using “&” to indicate to pull from the image list randomly *with* replacement.

Condition 3

Condition 3 contains three blocks, labeled 1-3 in column (C). You can see in column (D) that block_randomization for all three blocks are set to 1. This means that the order in which each of block 1-3 occurs within the time course of the study will be random. Each block contains a mix of trial types, showing the range of what you can do to customize your studies.

Block 1 begins with an instruction slide, which is followed by a mixture of type 3-6 trials. This shows that you can use as many trial types within a single block as you would like. Unique to Condition 3 is the use of column (U), which specifies an image location for the suppressed image. You can see that across all single-stimulus trials and all multi-stimulus trials, all location options are used (1-9). Remember, the location options depend on whether it is a single- or multi-stimulus trial type, so make sure you are familiar with those options when building your own study.

Block 2 is the same structure as Block 1, but now the option to set a time to max opacity (T) is used. Here you can see that the time to max opacity is set to 500.

Block 3 introduces some new features. You can see that all trials are still the basic trial type 3, but now all trials are “response” trials since we have entered response values into columns (O-R). Remember, simply by having values in those four columns, any trial can be a response trial. And, as you can see in trials 2-5, some of these response trials are multi-response trials, meaning all responses rendered by the participant within the Trial Duration will be recorded.

Condition 4

Condition 4 only contains one block. The point here is to show that you can mix as many types of trials as you would like within a single block. Here all 6 trial types are contained within a single block. (Note, we do not recommend doing this in real life, but there may be a situation in which you find the need to have many trial types within a single block and you can do so).

Section 9: Other Features

9.1 Running through the Command Line

You can run CFS-VR directly through the command line in windows. First (in the command prompt window), you will need to navigate your command line to the directory of CFS-VR. This will most wherever you have copied the CFS-VR main folder. In the below example, that folder was unzipped and pasted into the Program Files (x86) folder. The program can be started by running the command “CFS-VR.exe.”

To launch the program directly into a study, additional arguments must be added to this command in this order. The first three are necessary and the last two are optional:

- 1st** (necessary): The file location of your study file.
 - Example: “C:\users\username\Desktop\UserTemplateCFSVR.csv”
- 2nd** (necessary): The Participant ID.
 - Example: “Sam01”
- 3rd** (necessary): Their eye dominance in lowercase.
 - Example: “right”
- 4th** (optional): Output file location.
 - Example: “C:\users\username\Desktop\”
- 5th** (optional): The order of your conditions.
 - Example: “312”

An example of a command all together would be something like the below where the first two lines navigate to the correct folder and the last entry begins a study:

```
C:\>cd "Program Files (x86)"  
C:\Program Files (x86)>cd CFS-VR  
C:\Program Files (x86)\CFS-VR>CFS-VR.exe C:\users\username\Desktop\UserTemplateCFSVR.csv Sam01 right  
C:\users\username\Desktop\ 312
```

Good luck with your research! Please remember that this is free software, we are not a software company, and we have no budget or personnel for technical support!