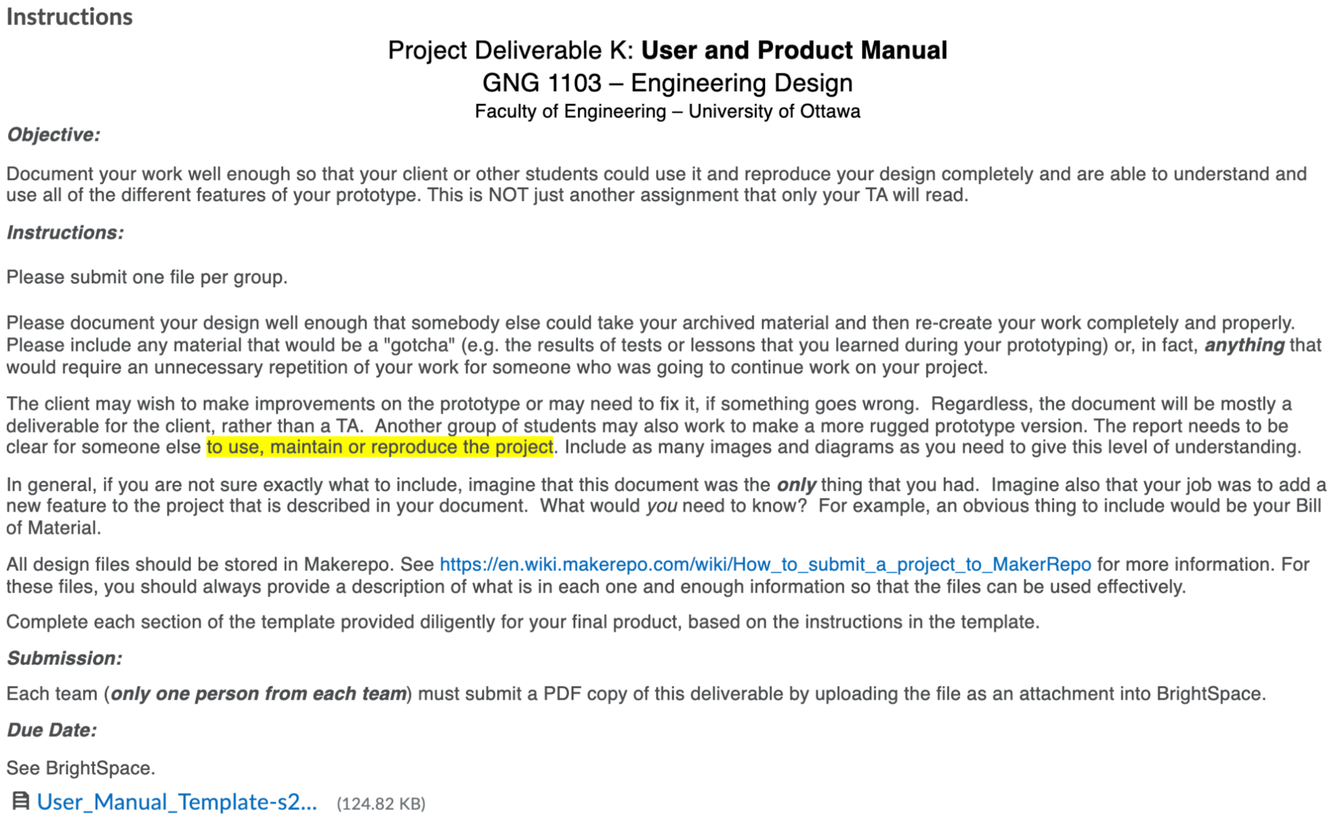
Project Deliverable K: Product Manual

Tyler Heimbecker #300232288
Emily Thompson #300238169
Nicolas Becerril #0300231477
Riley Johnston #300119395

December 8th, 2021

Abstract

This is Deliverable K: Product Manual for the Ross Video Dashboard group project. The document details the steps to recreate our final prototype, as well as a detailed description of how to use the panel once you have access to it. The main purpose is to outline the final designs for this step in the group project if anyone wants to replicate or use the final product.



Instructions

Project Deliverable K: **User and Product Manual**
GNG 1103 – Engineering Design
Faculty of Engineering – University of Ottawa

Objective:

Document your work well enough so that your client or other students could use it and reproduce your design completely and are able to understand and use all of the different features of your prototype. This is NOT just another assignment that only your TA will read.

Instructions:

Please submit one file per group.

Please document your design well enough that somebody else could take your archived material and then re-create your work completely and properly. Please include any material that would be a "gotcha" (e.g. the results of tests or lessons that you learned during your prototyping) or, in fact, ***anything*** that would require an unnecessary repetition of your work for someone who was going to continue work on your project.

The client may wish to make improvements on the prototype or may need to fix it, if something goes wrong. Regardless, the document will be mostly a deliverable for the client, rather than a TA. Another group of students may also work to make a more rugged prototype version. The report needs to be clear for someone else to use, maintain or reproduce the project. Include as many images and diagrams as you need to give this level of understanding.

In general, if you are not sure exactly what to include, imagine that this document was the ***only*** thing that you had. Imagine also that your job was to add a new feature to the project that is described in your document. What would *you* need to know? For example, an obvious thing to include would be your Bill of Material.

All design files should be stored in Makerepo. See https://en.wiki.makerepo.com/wiki/How_to_submit_a_project_to_MakerRepo for more information. For these files, you should always provide a description of what is in each one and enough information so that the files can be used effectively.

Complete each section of the template provided diligently for your final product, based on the instructions in the template.

Submission:

Each team (***only one person from each team***) must submit a PDF copy of this deliverable by uploading the file as an attachment into BrightSpace.

Due Date:

See BrightSpace.

User_Manual_Template-s2... (124.82 KB)

Table of Contents

Introduction--- 4

Using the Panel-- 4 - 9
- **1.1 Crowd Prompts**-- 4
- **1.2 Sponsor Bugs**--- 4 - 6
- **1.3 Standings**-- 6 - 8
- **1.4 Fan Birthdays**--- 8

Constructing the Panel-- 9 - 32
- **2.1 Crowd Prompts**-- 9 - 11
- **2.2 Sponsor Bugs**--- 12 - 21
- **2.3 Standings**--- 21 - 27
- **2.4 Fan Birthdays**--- 27 - 32

Integrating all Panels--- 33

MakerRepo-- 33

Introduction

Throughout the process of this project, our group created rough drafts of each subsection, executed our ideas on Dashboard, and presented our final ideas and product.

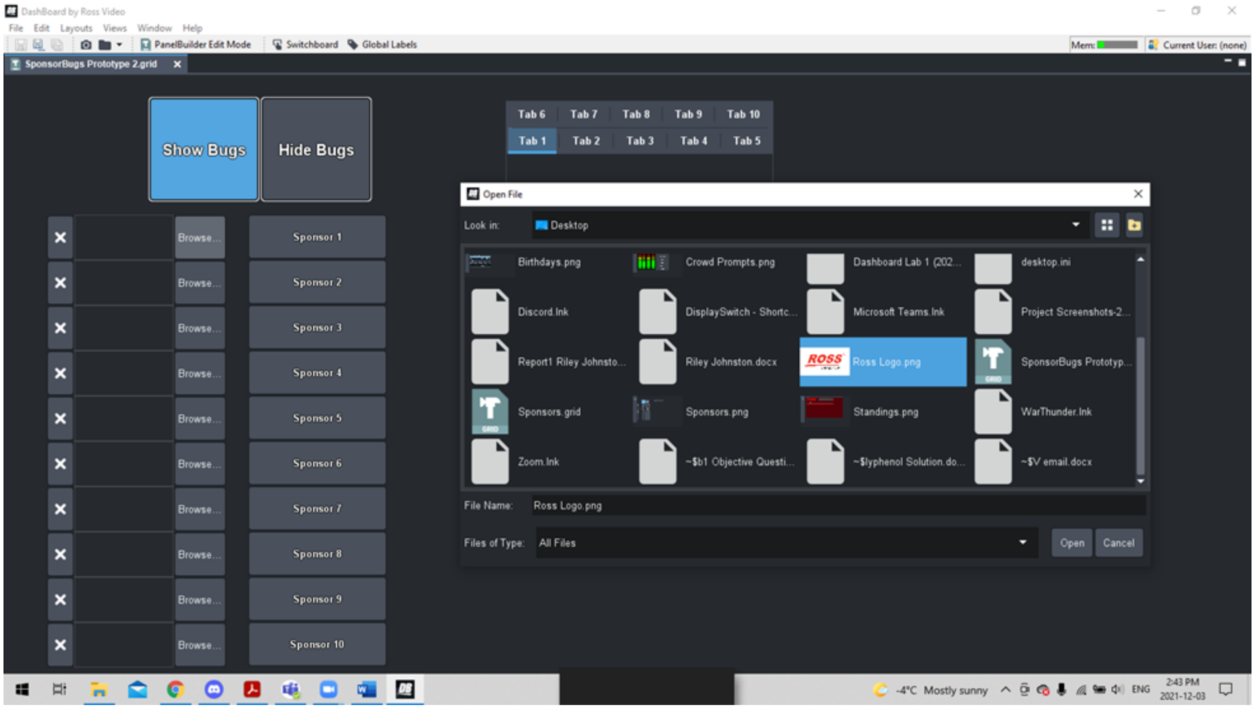
To Use the Panel

To Use Crowd Prompts Panel

The crowd prompts panel is debatably the most simple panel to use. All you must do is identify the correct take id's of images you wish to display within each of the buttons source code and you should be good from there. The buttons are programmed in a simple yet effecting in and out arrangement meaning the user can both send out an image to display, and call it back in case of an accidental input or as a hard shut off.

To Use Sponsor Bugs Panel

To upload a sponsor photo, click one of the “Browse” buttons. This will open the file select widget, where you can select a photo to be displayed. Here we picked “Ross Logo.png”. Once you click the file, press the “Open” button at the bottom right of the widget.

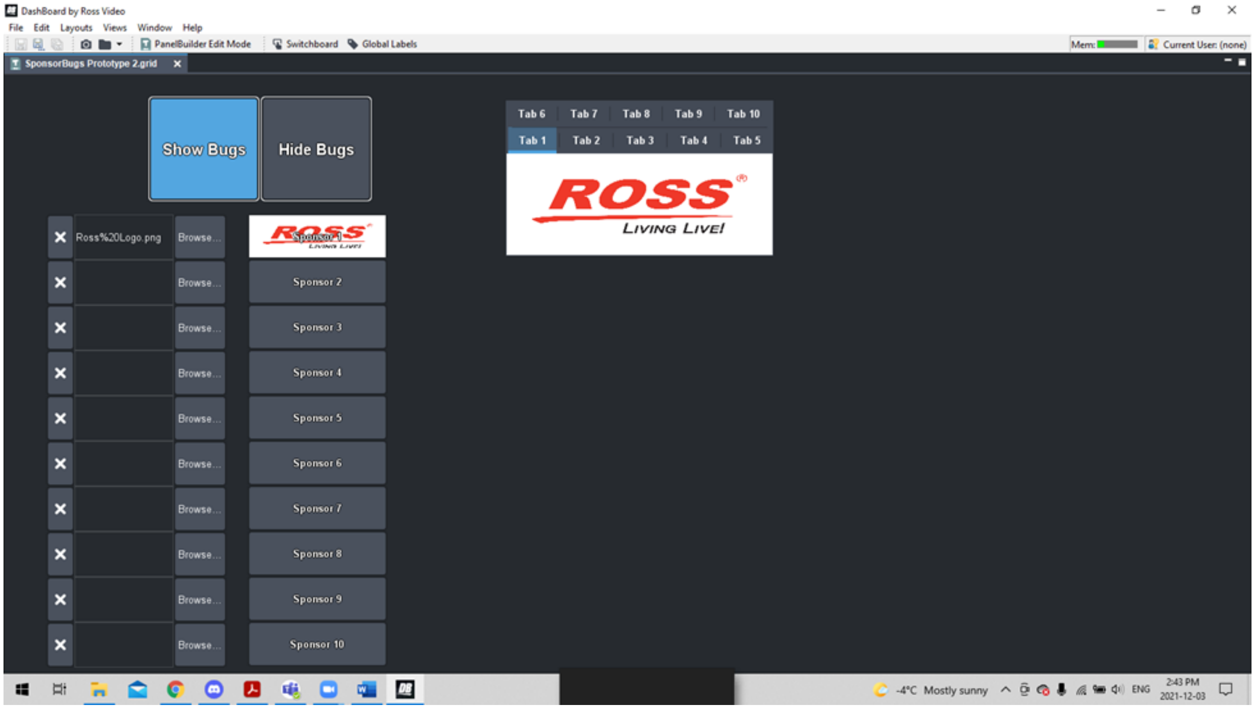


Look in: Desktop

File Name: Ross Logo.png

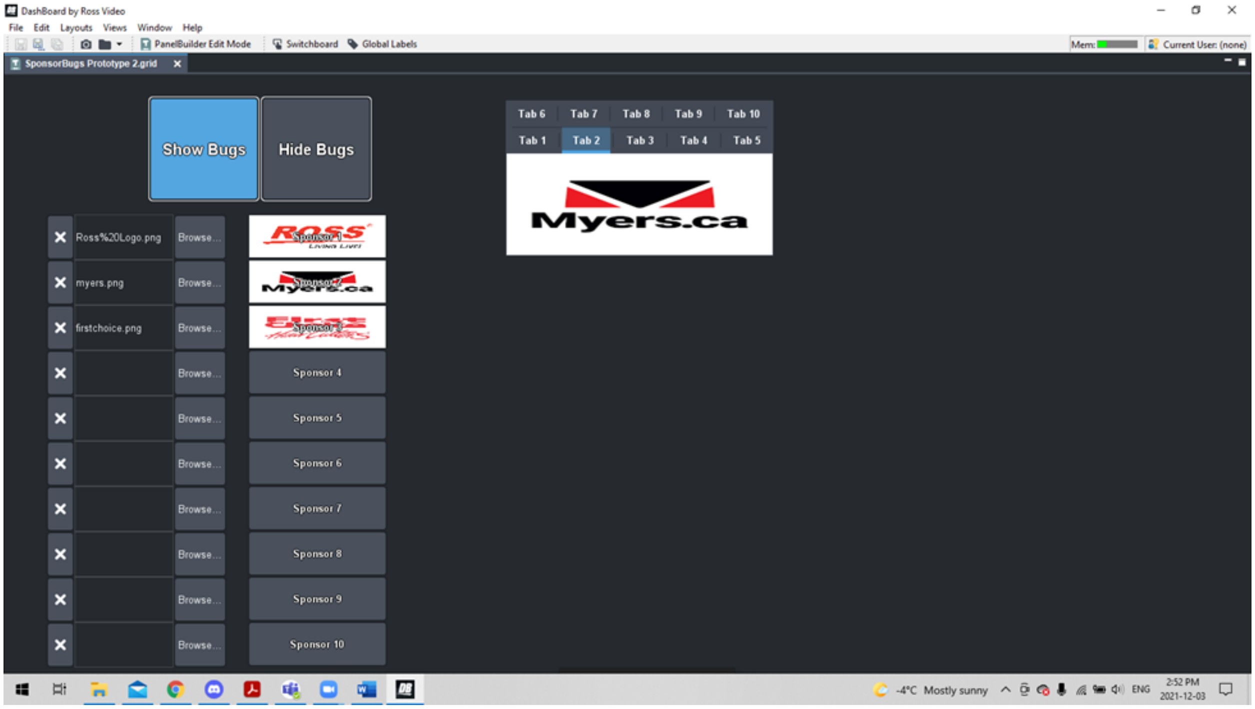
Files of Type: All Files

Now we can see that the “Sponsor 1” button is displaying the sponsor logo, as well as tab 2.



Ross%20Logo.png

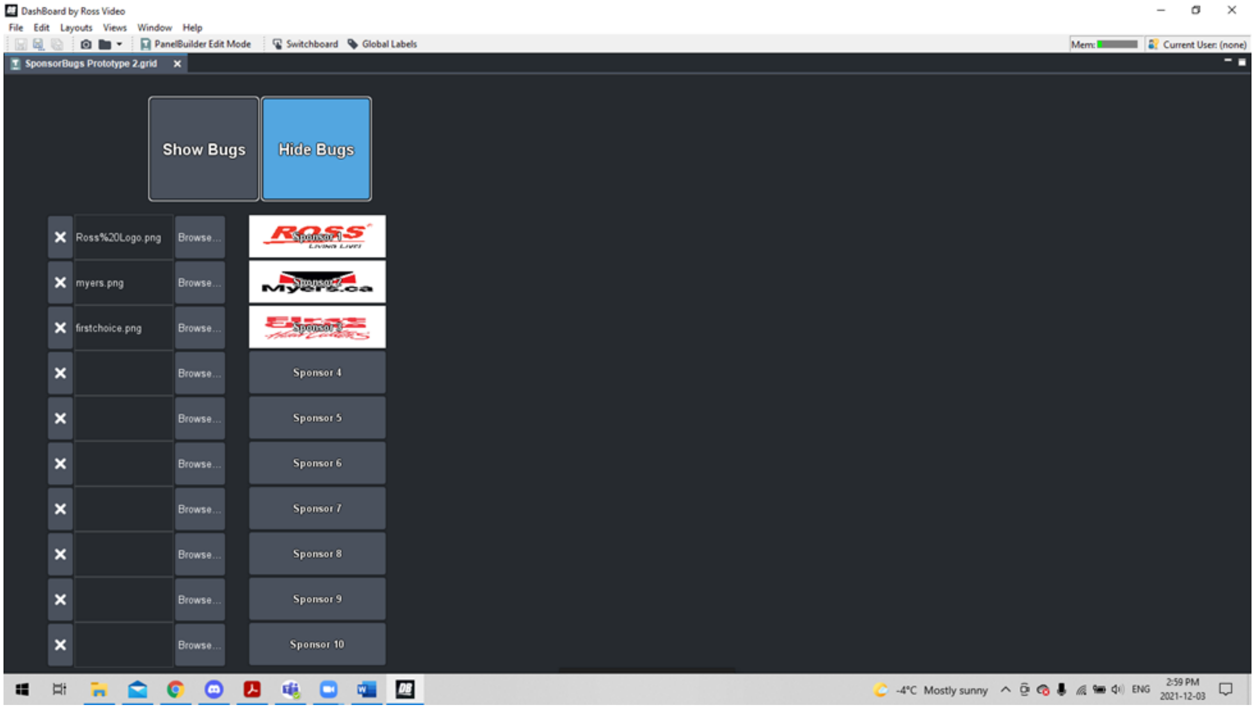
Let’s repeat these steps for a few more sponsors by using the Browse buttons…



Ross%20Logo.png
myers.png
firstchoice.png

We now have a second and third sponsor photo. To switch which one is displayed on the output tabs in the upper right corner, we simply click the buttons with the corresponding sponsor photos in them. Here the “Sponsor 2” button with the Myers.ca photo was clicked, and we can see that the Myers logo is being displayed in the output tabs.

Finally, if we want to stop showing the sponsor photos in the output tabs, we can click the “Hide Bugs” button at the top left of the panel. This will hide the output tabs. To toggle them back on again, click “Show Bugs”.

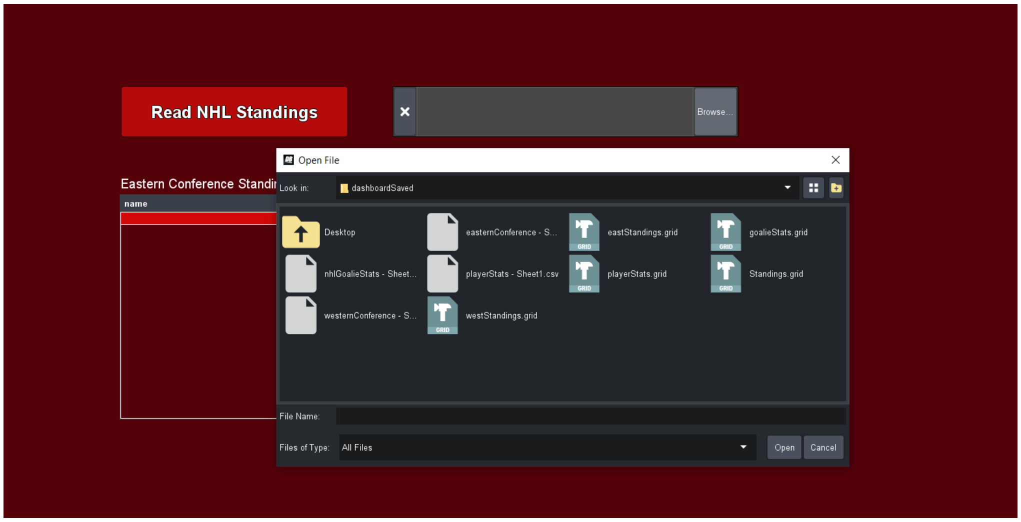


Ross%20Logo.png
myers.png
firstchoice.png

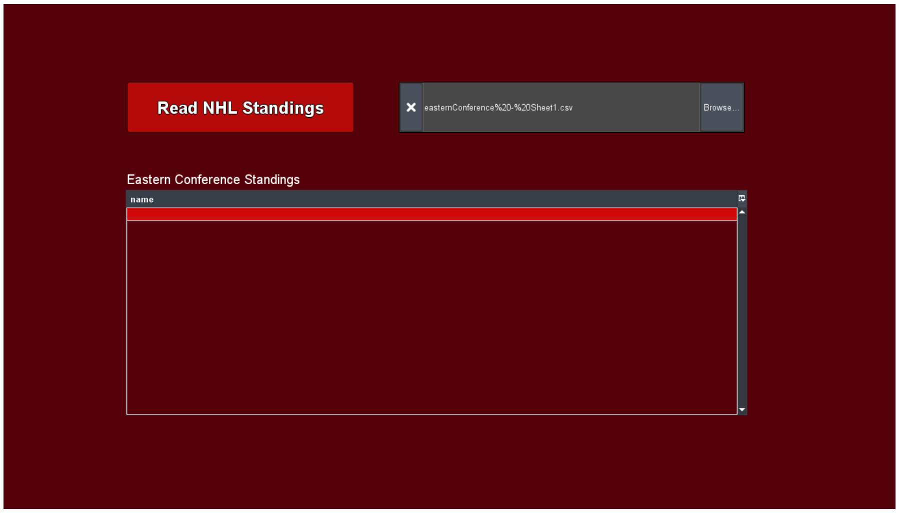
To Use Standings Panel

After your panel is built, the steps to use the standings panel are very simple.

1. First Click on the browse option. This will display all of your files, to select an appropriate file to display

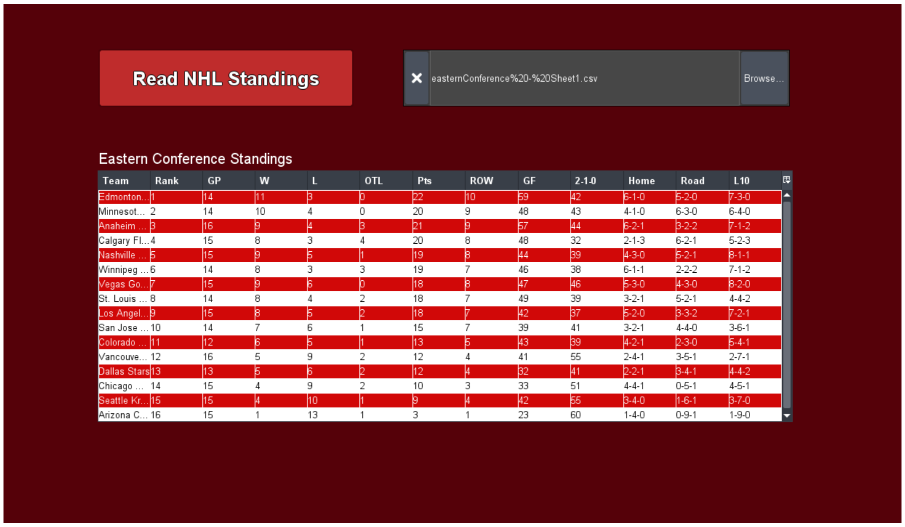


2. Next, select a file, appropriate to the panel.



easternConference%20-%20Sheet1.csv

3. Finally, click the “Read NHL Standings” button, and the information will be displayed to you in a table format.

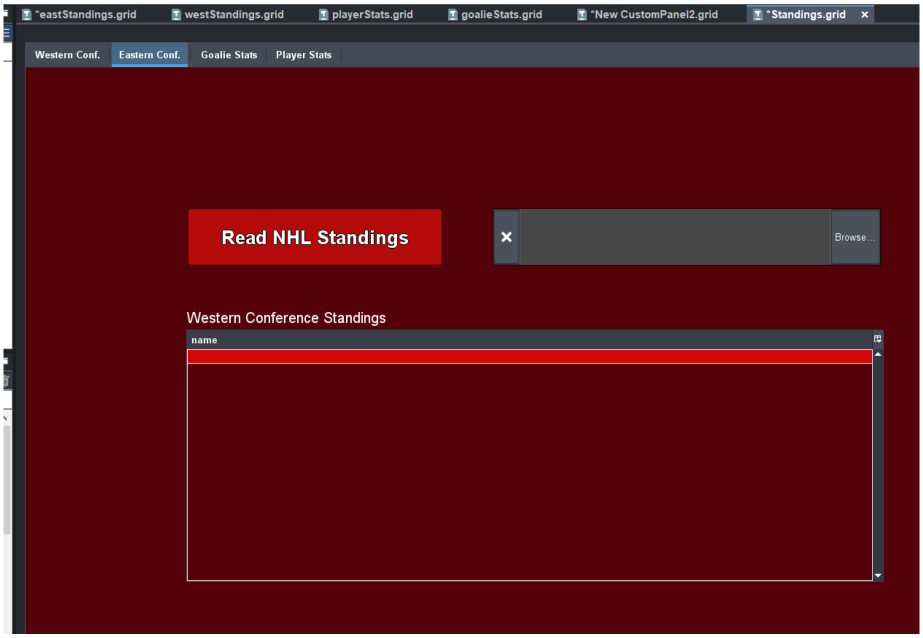


easternConference%20-%20Sheet1.csv

Eastern Conference Standings

Team	Rank	GP	W	L	OTL	Pts	ROW	GF	2-1-0	Home	Road	L10
Edmonton...	1	14	11	3	0	22	10	59	42	6-1-0	5-2-0	7-3-0
Minnesot...	2	14	10	4	0	20	9	48	43	4-1-0	6-3-0	6-4-0
Anaheim ...	3	16	9	4	3	21	9	57	44	6-2-1	3-2-2	7-1-2
Calgary Fl...	4	15	8	3	4	20	8	48	32	2-1-3	6-2-1	5-2-3
Nashville ...	5	15	9	5	1	19	8	44	39	4-3-0	5-2-1	8-1-1
Winnipeg ...	6	14	8	3	3	19	7	46	38	6-1-1	2-2-2	7-1-2
Vegas Go...	7	15	9	6	0	18	8	47	46	5-3-0	4-3-0	8-2-0
St. Louis ...	8	14	8	4	2	18	7	49	39	3-2-1	5-2-1	4-4-2
Los Angel...	9	15	8	5	2	18	7	42	37	5-2-0	3-3-2	7-2-1
San Jose ...	10	14	7	6	1	15	7	39	41	3-2-1	4-4-0	3-6-1
Colorado ...	11	12	6	5	1	13	5	43	39	4-2-1	2-3-0	5-4-1
Vancouve...	12	16	5	9	2	12	4	41	55	2-4-1	3-5-1	2-7-1
Dallas Stars	13	13	5	6	2	12	4	32	41	2-2-1	3-4-1	4-4-2
Chicago ...	14	15	4	9	2	10	3	33	51	4-4-1	0-5-1	4-5-1
Seattle Kr...	15	15	4	10	1	9	4	42	55	3-4-0	1-6-1	3-7-0
Arizona C...	16	15	1	13	1	3	1	23	60	1-4-0	0-9-1	1-9-0

4. You can switch in between the tabs in order to change the standings you are viewing, quickly.



To Use Fan Birthday Panel

1. Once you have access to this panel, it is very effective and easy to use. To edit the birthday button, go to the birthday config tab and add the required information: this includes the name, age, message, and image of the fan whose birthday it is.

2. You must manually input all this information, but you can copy and paste from an external source to avoid errors. Use the spinner to adjust the age of the fan. Browse for an image and add it to the "Image" section.

3. Repeat for as many birthdays as you wish to display. The data table tab will display all the necessary information.

4. Set the time duration by clicking the button to do so. Change the time for each birthday in the source code.

5. Edit what you want the main message to say. In this panel, we chose to say "Today is a special day for…".

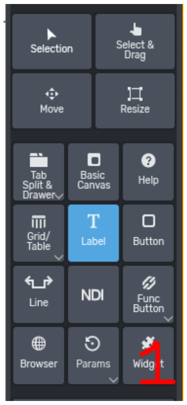
6. When displaying the birthdays you've selected, hit "Intro Display" and cue any music you want in the arena. Hit "Display All" and this will start the timer.

7. If you need to quickly add or remove a birthday, hit the button that corresponds to that birthday and the button will perform the task for you. Another option is to hit "stop" on the timer, and this will suspend all actions.

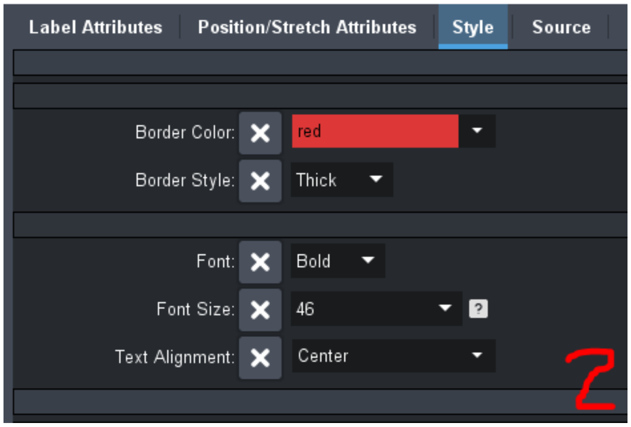
To Build the Panel

To Build Crowd Prompts Panel

1. Navigate to a new or existing empty dashboard panel
2. Create a title for the panel by selecting PanelBuilder Edit mode (or press Ctrl +G), then selecting the label button in the image below (image 1)



3. Select a space on your panel where you wish to add the label and left-click and drag until the label is the desired size (you should see a green outlined box as you drag)
4. When you let go of the left click a prompt will pop up asking what the desired name, ID, text alignment, and header. Fill out the name as “Crowd Prompts”, leave ID blank, select a center text alignment, and check off the box that says header.
5. Navigate to the Panel editor on the left of your screen and select “Select & Drag” and double left click on the label box you have just created.
6. At the top of your screen, you should see a button that says “Style”. Click this button and fill out the data to match image 2.



Border Color: red

Border Style: Thick

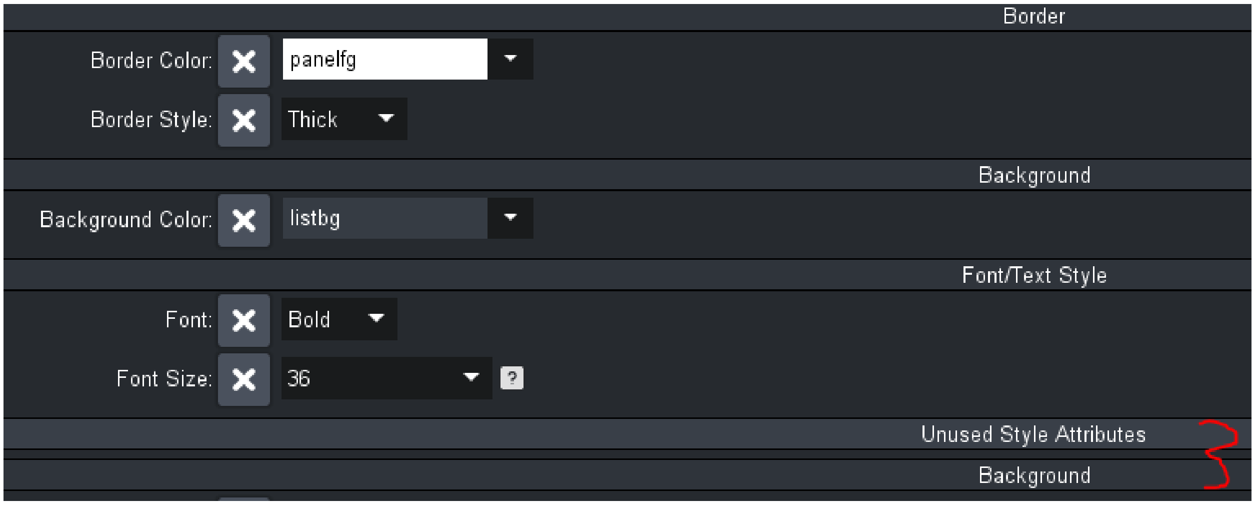
Font: Bold

Font Size: 46

Text Alignment: Center

7. Create a simple grid by navigating to the edit tab on the left, clicking grid/table, then a simple grid. Left-click on the canvas and drag to the desired size. Click the override default box on both the row and column tabs then change rows to 6 and columns to 1.
8. Create a button by clicking the ‘button’ button on the edit tab on the left and then left-clicking the new grid. Change the name to ‘Make Some Noise’, the id to ‘Noise_’, and the type to radio.

9. Create another button by following the same steps but change the name to ‘Noise Meter’ and ID to ‘_meter’
10. Create another button by following the same steps but change the name to ‘Clap’ and ID to ‘Clap_’
11. Create another button by following the same steps but change the name to ‘Goal’ and ID to ‘Goal_’
12. Create another button by following the same steps but change the name to ‘Penalty’ and ID to ‘Penalty_’
13. Create another button by following the same steps but change the name to ‘Replay’ and ID to ‘Replay_’
14. Navigate to the Style tab in each button and change their styles to match image 3.



Border

Border Color: panelfg

Border Style: Thick

Background

Background Color: listbg

Font/Text Style

Font: Bold

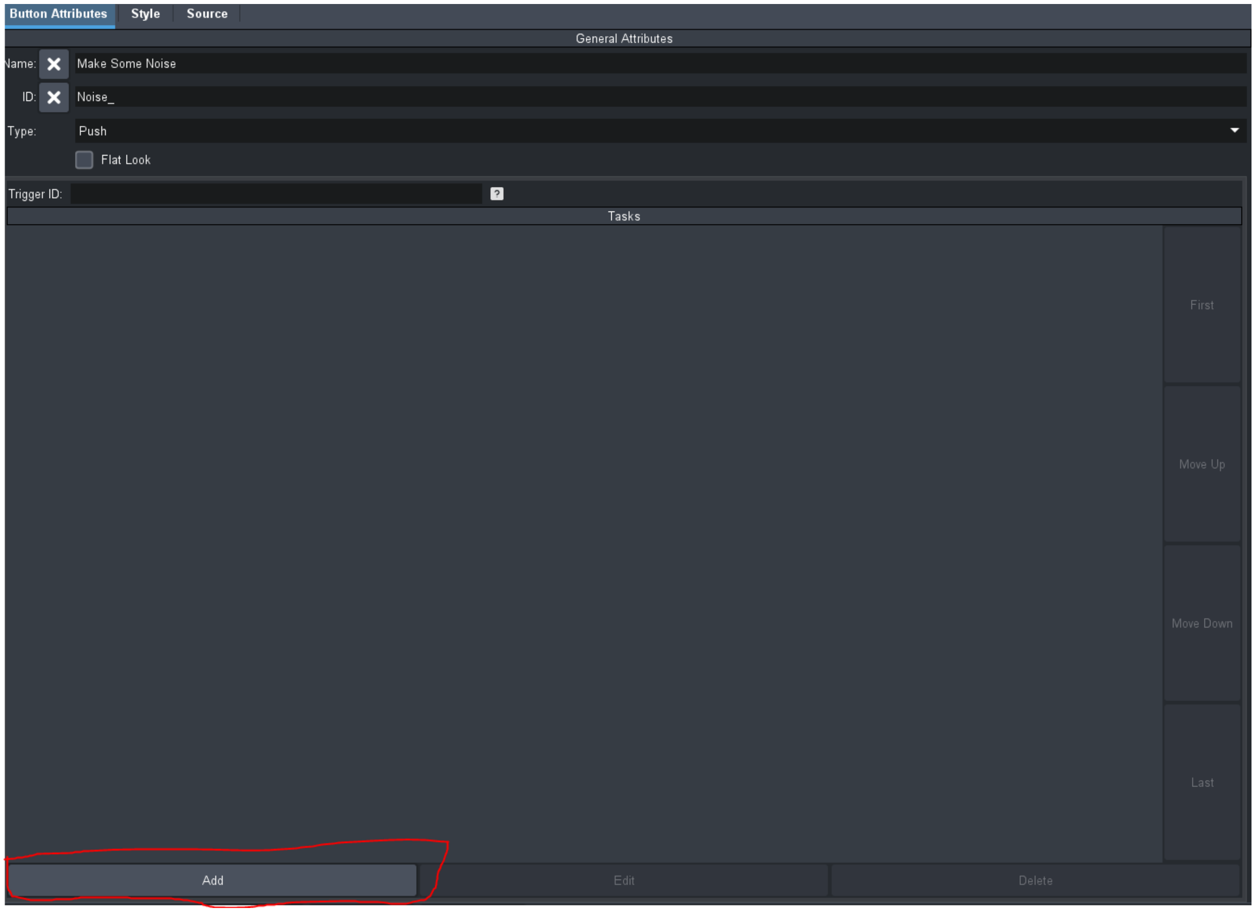
Font Size: 36

Unused Style Attributes

Background

3

15. Navigate to the button attributes tab for each of these buttons and press the ‘add’ button on the bottom left.



Name: Make Some Noise

ID: Noise_

Type: Push

☐ Flat Look

Trigger ID:

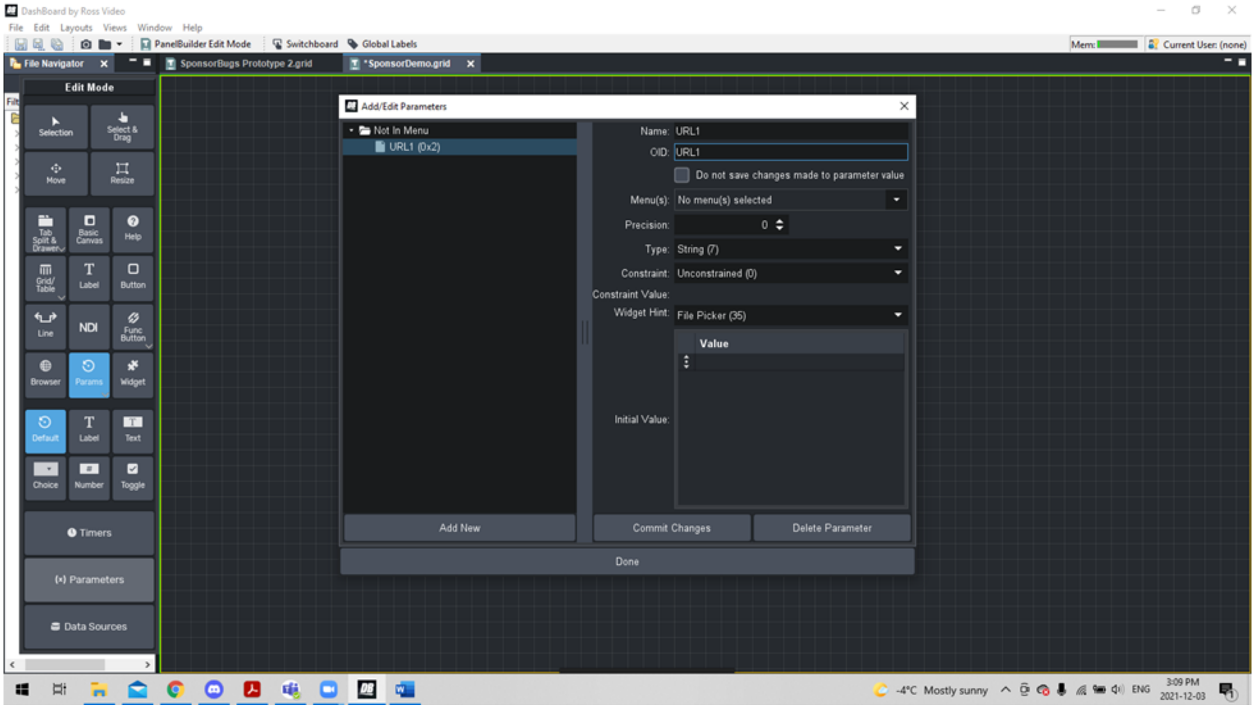
16. Add a take function to each button by navigating to the drop-down menu on the left under expression. Click API, then click ‘Take’. Do not change the buffer or layer values, but change the take id to the desired value for the game. Repeat this step for each of the buttons while changing the take ids to each of the desired values.
17. After this, create a second simple grid with 1 column and 6 rows. Fill this column with 6 buttons, to make it less complicated keep the id of each button the same as its name.
18. Create the first button as ‘Noise Out’. Create the second button as ‘Noisemeter Out’. Create the third button as ‘Clap Out’. Create the fourth button as ‘Goal Out’. Create the fifth button as ‘Penalty Out’. Create the last button as ‘Replay Out’.
19. Change the style of each of these new buttons to have a red fill, thick border, and font size 30.

<u>To Build Sponsor Bugs Panel</u>

Part 1. Creating the File Pickers

Start by creating a new custom panel file on Dashboard, and enter PanelBuilder Edit mode by clicking the button at the top left of the screen.

Create a new parameter and give it the name and OID of URL1. It should have the same properties as shown below, and its default widget hint should be File Picker. Click “Done.”



Name: URL1

OID: URL1

☐ Do not save changes made to parameter value

Menu(s): No menu(s) selected

Precision: 0

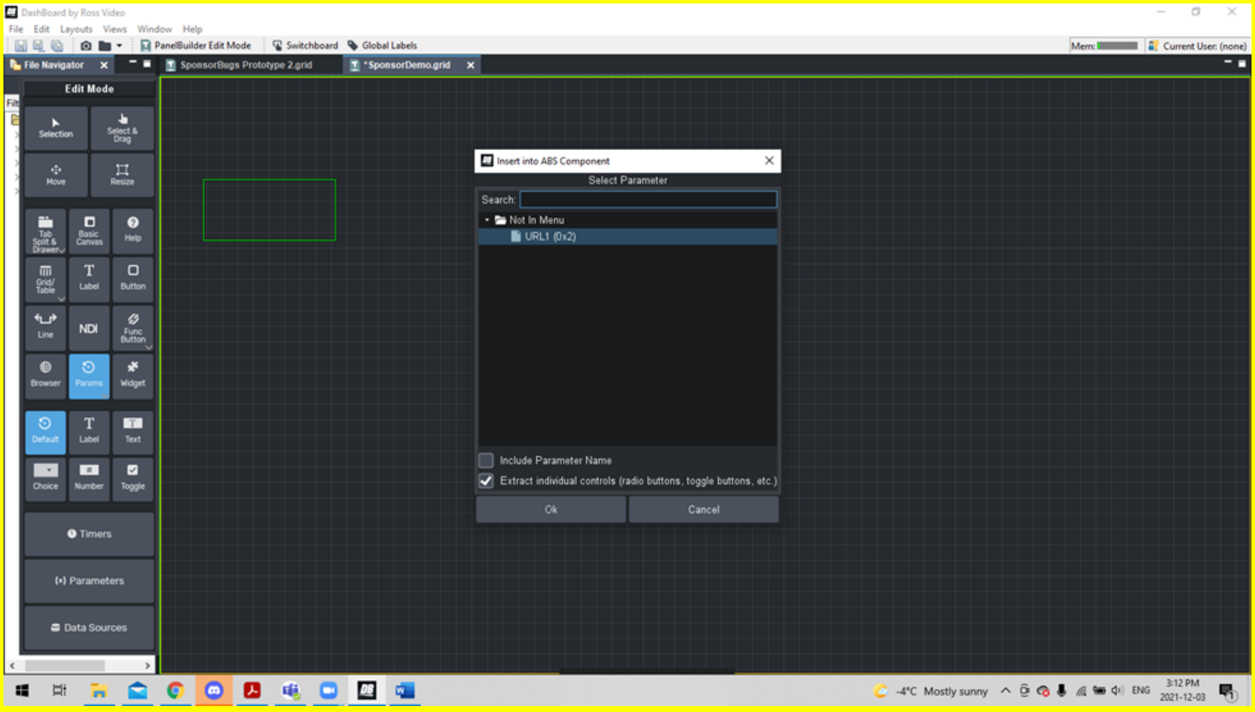
Type: String (7)

Constraint: Unconstrained (0)

Constraint Value:

Widget Hint: File Picker (35)

Initial Value:



Insert into ABS Component

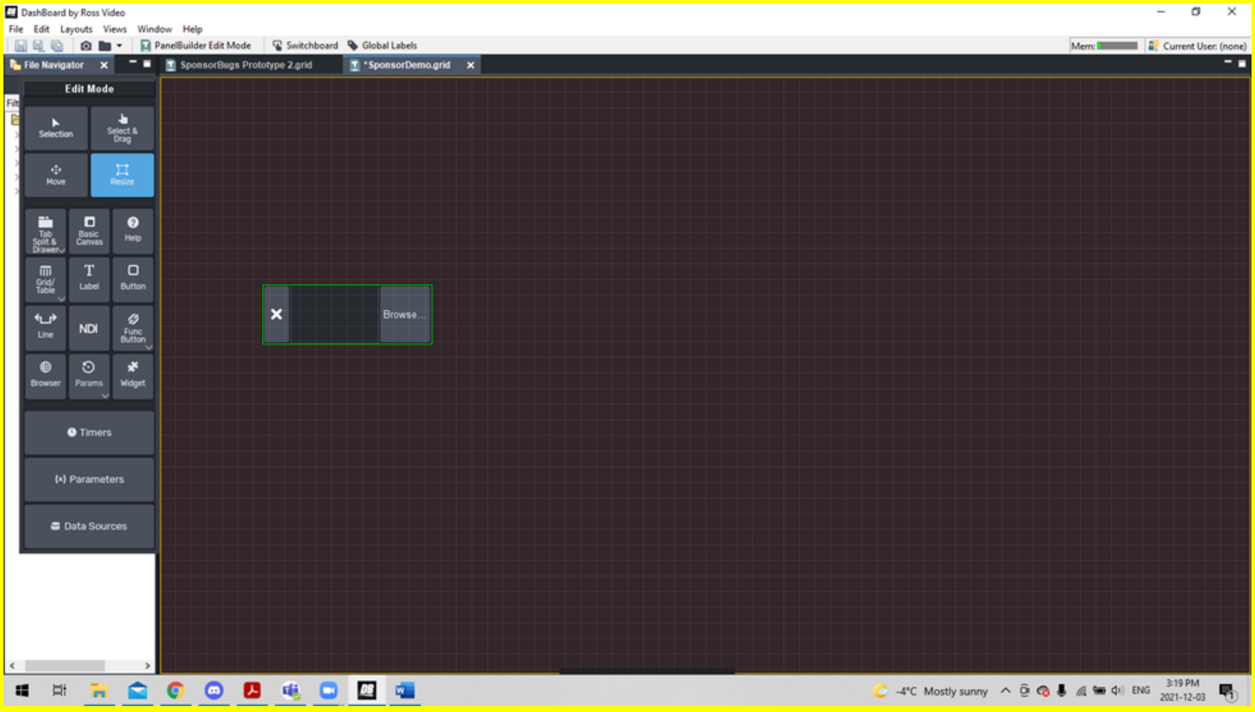
Select Parameter

Search:

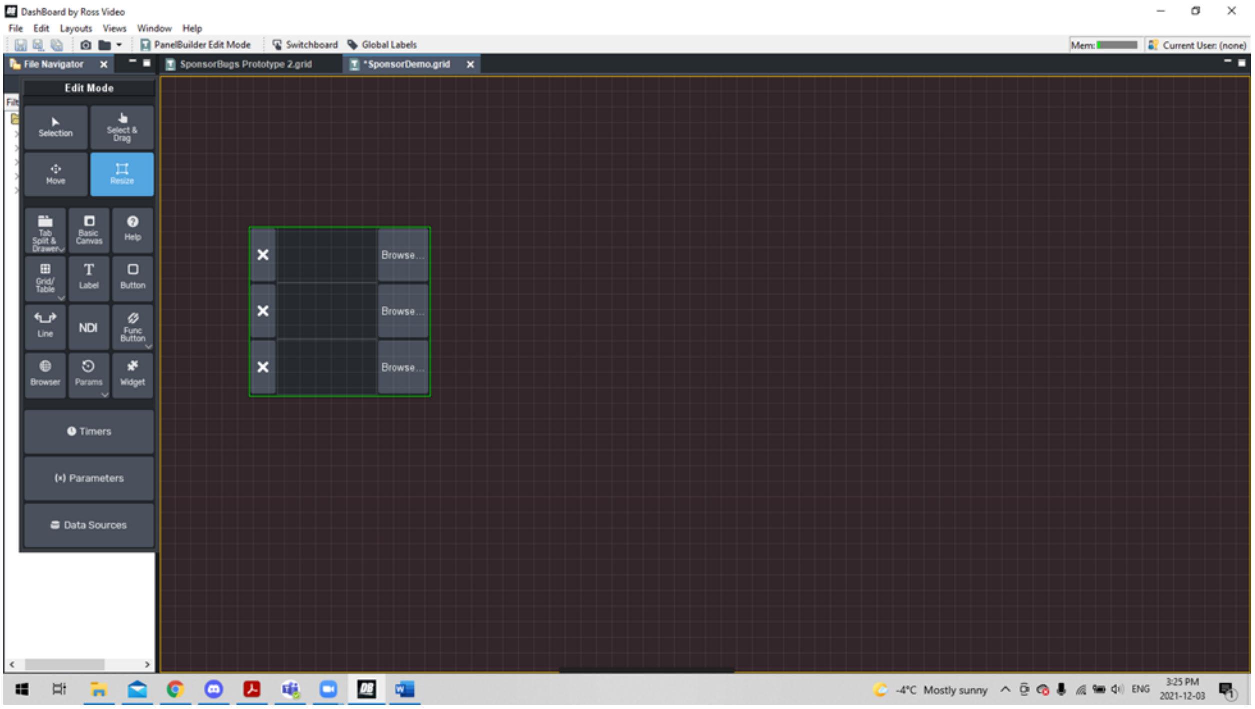
- Not In Menu
  - URL1 (0x2)

☐ Include Parameter Name

☑ Extract individual controls (radio buttons, toggle buttons, etc.)

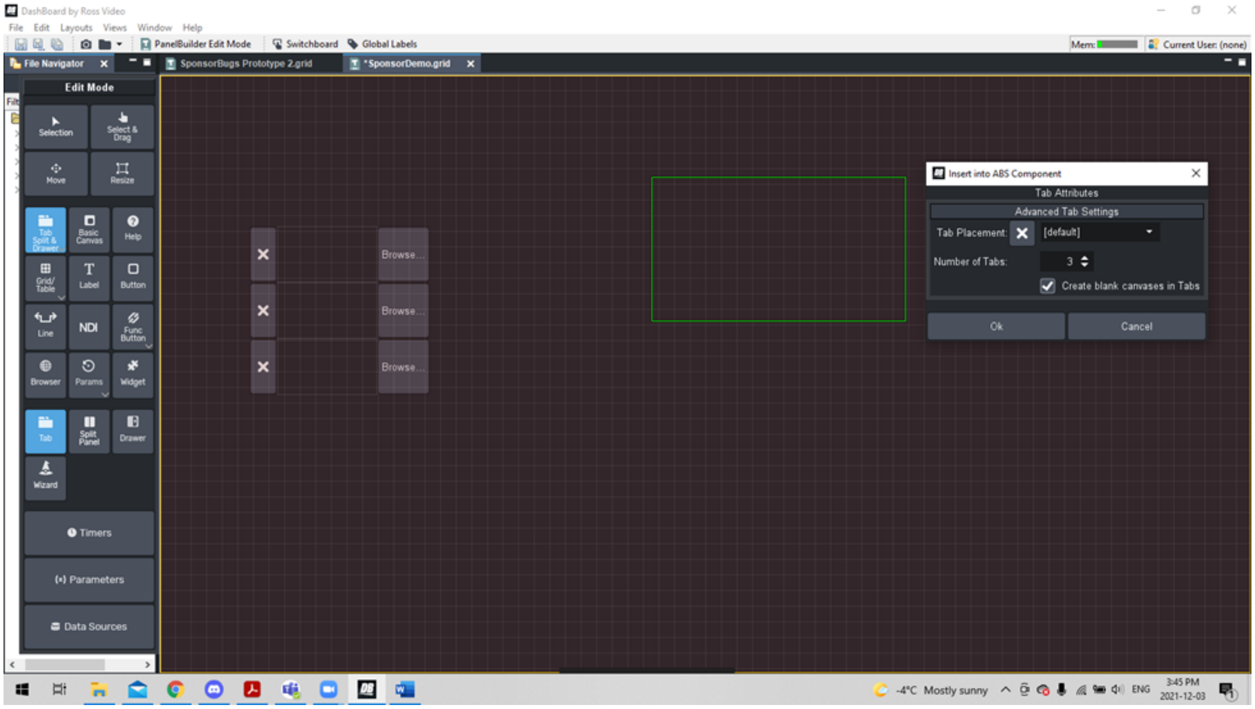


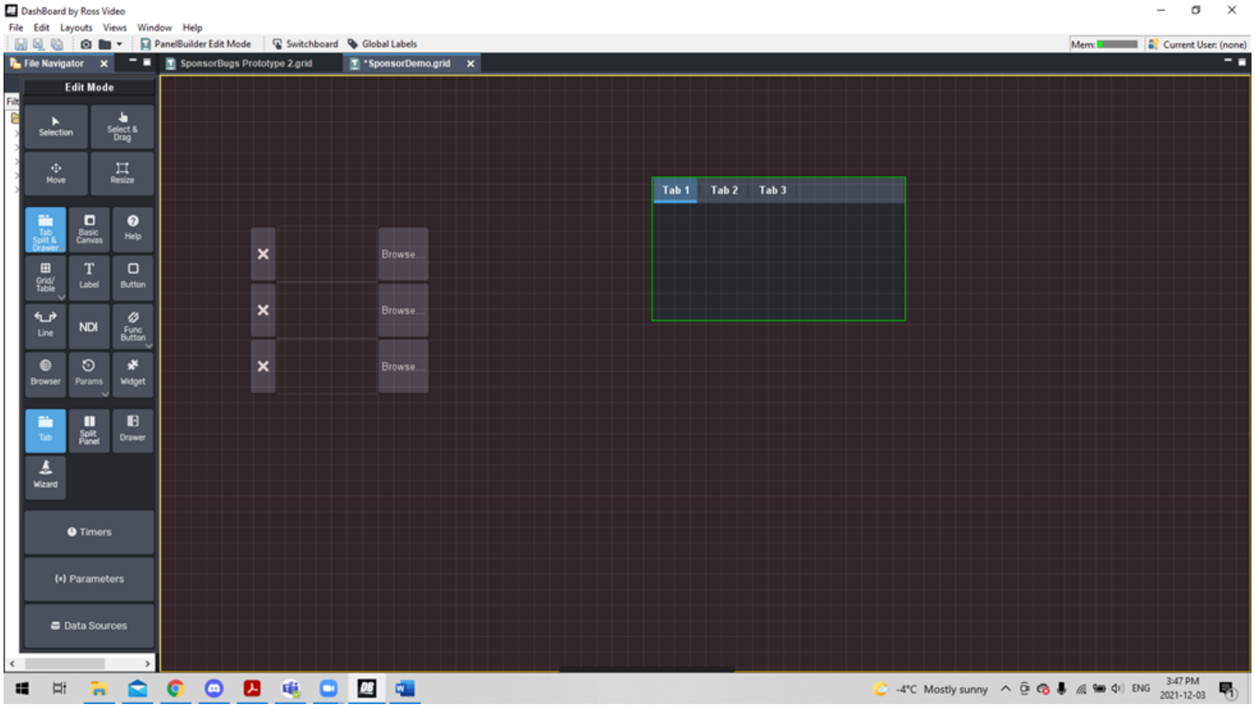
We will need a file picker for each sponsor photo we want to include. For the purposes of this tutorial, we will do 3 file pickers. Repeat the previous steps, giving the new parameters names and OIDs of URL2 and URL3. Place the 3 file pickers on the panel as shown below.



Part 2. Creating the Display Tabs

In the toolbar, click Tab&Split Drawer, and then click Tab. Drag and drop on the panel where you want the display tabs to be, and a pop up window will appear. The number of tabs will need to be the same as the number of file pickers, so we will set this to 3. Make sure Tab Placement is set to Default and that “Create Blank Canvases in Tabs” is checked. Press OK.

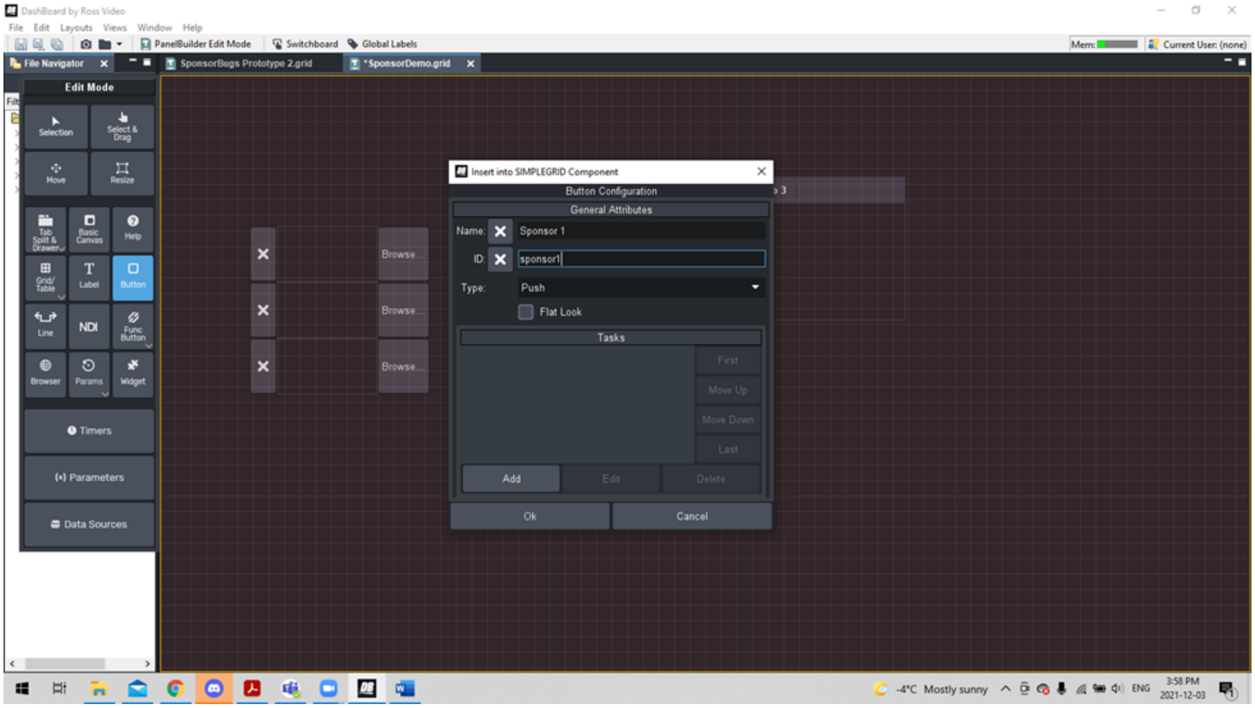

Insert into ABS Component
Tab Attributes
Advanced Tab Settings
Tab Placement: [default]
Number of Tabs: 3
☑ Create blank canvases in Tabs



Double click each tab and make the ID the same as its name. Finally, click the border of the tabs and give the canvas the id “sponsor_display”. (This is the id for the group that all tabs are members of).

Part 3. Creating Sponsor Buttons

We need to create a button for each of the file pickers so that we can select which sponsor photo is displayed in the tabs. Click the Buttons tool in the toolbar and drag and drop a button to the right of each of the file pickers. Give the buttons names “Sponsor x” and ID’s of “sponsorx”, where x is the numbers 1 through 3.



Insert into SIMPLEGRID Component

Button Configuration

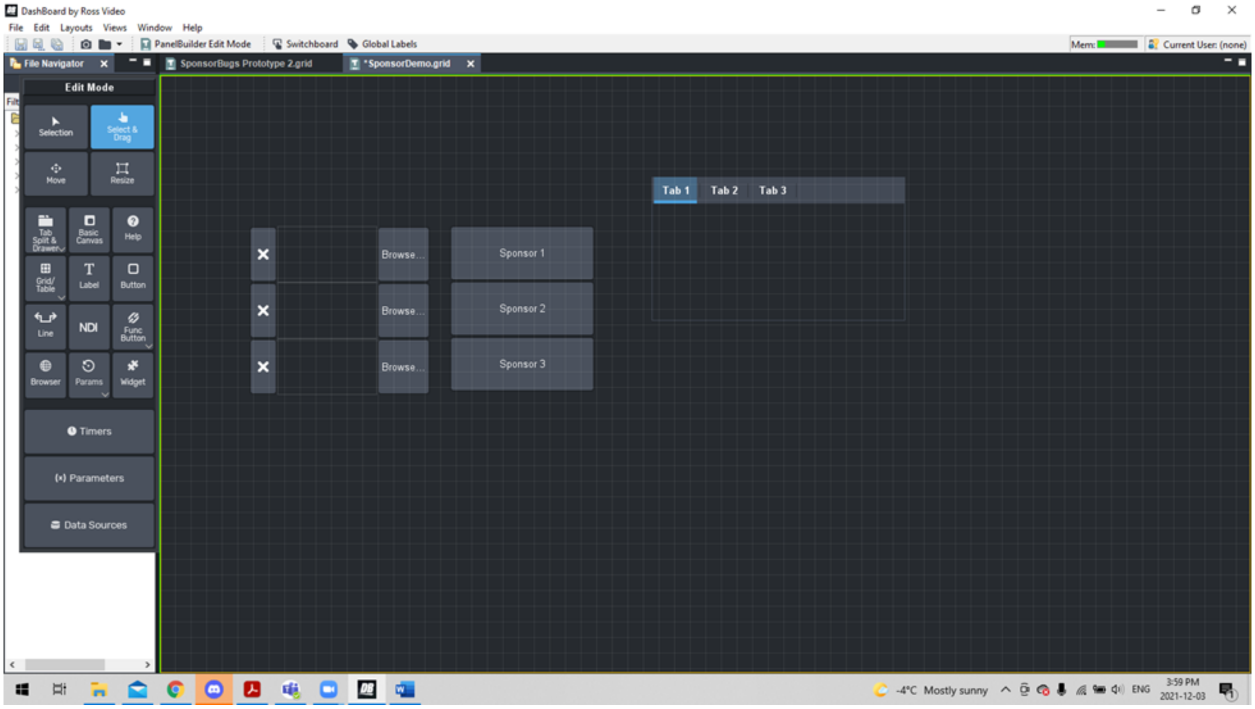
General Attributes

Name: Sponsor 1

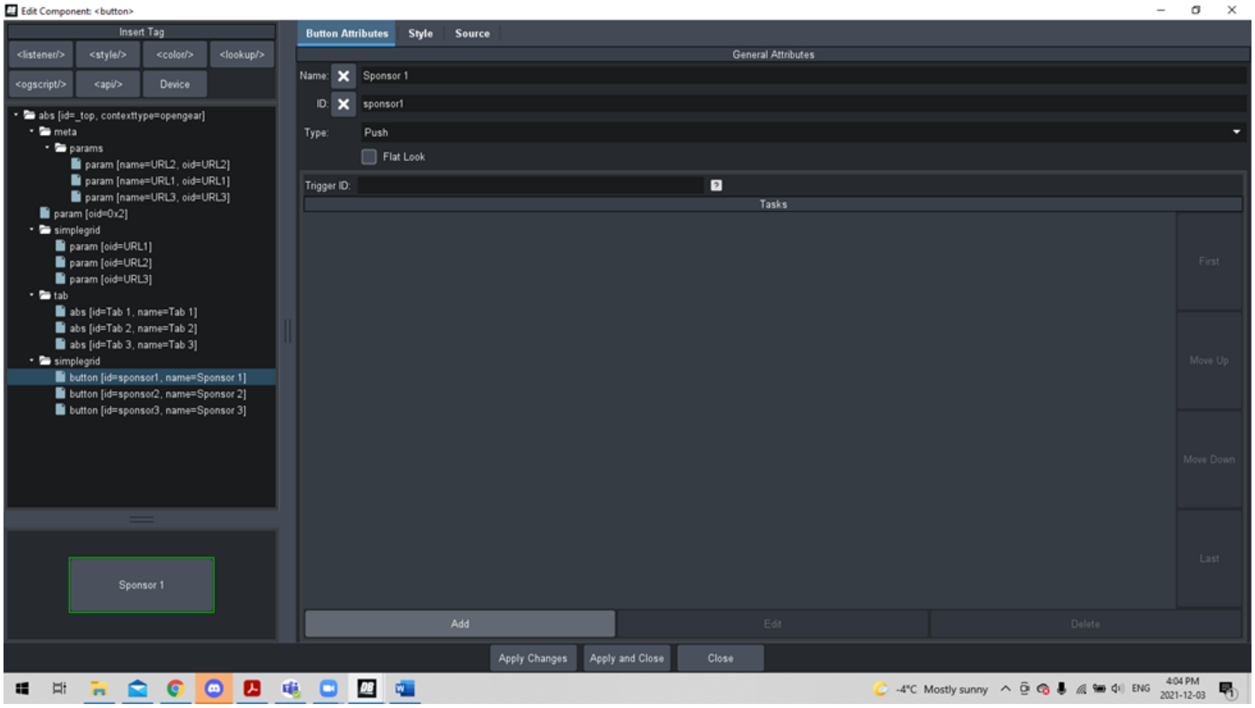
ID: sponsor1

Type: Push

☐ Flat Look



Now we can add tasks to the buttons. Using the Select and Drag tool, double click the Sponsor 1 button. The window below will appear.

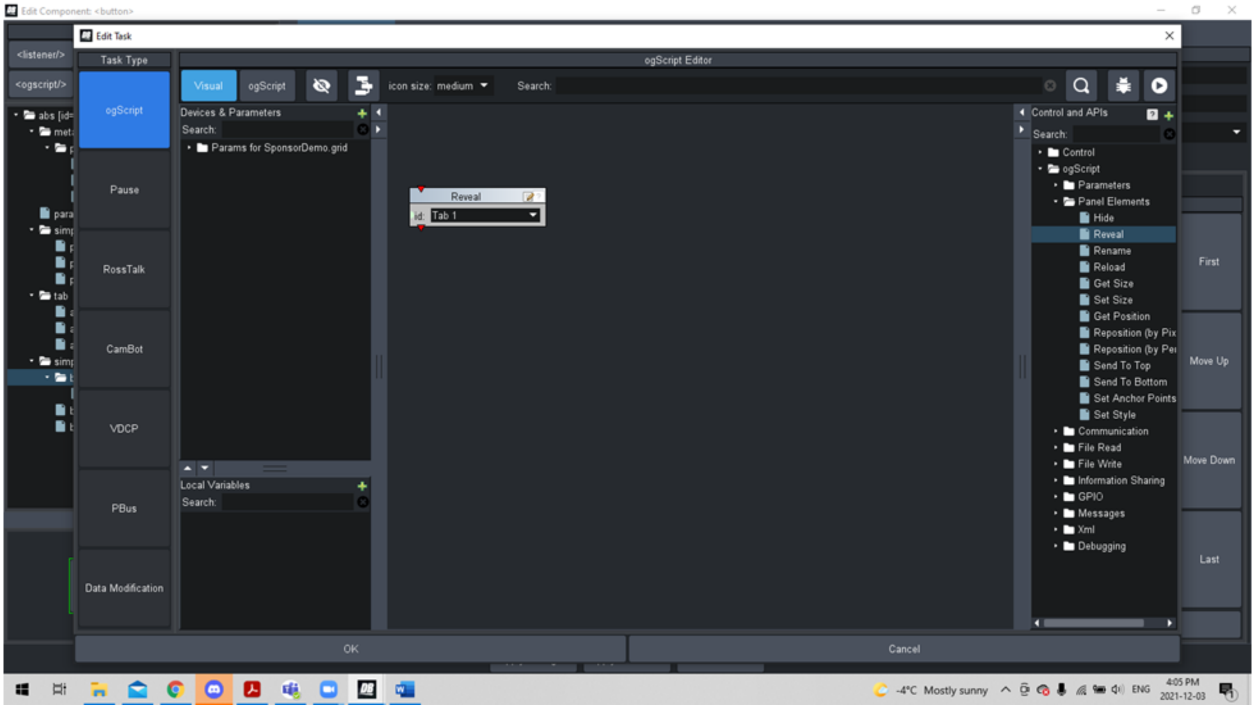


Name: Sponsor 1

ID: sponsor1

Type: Push

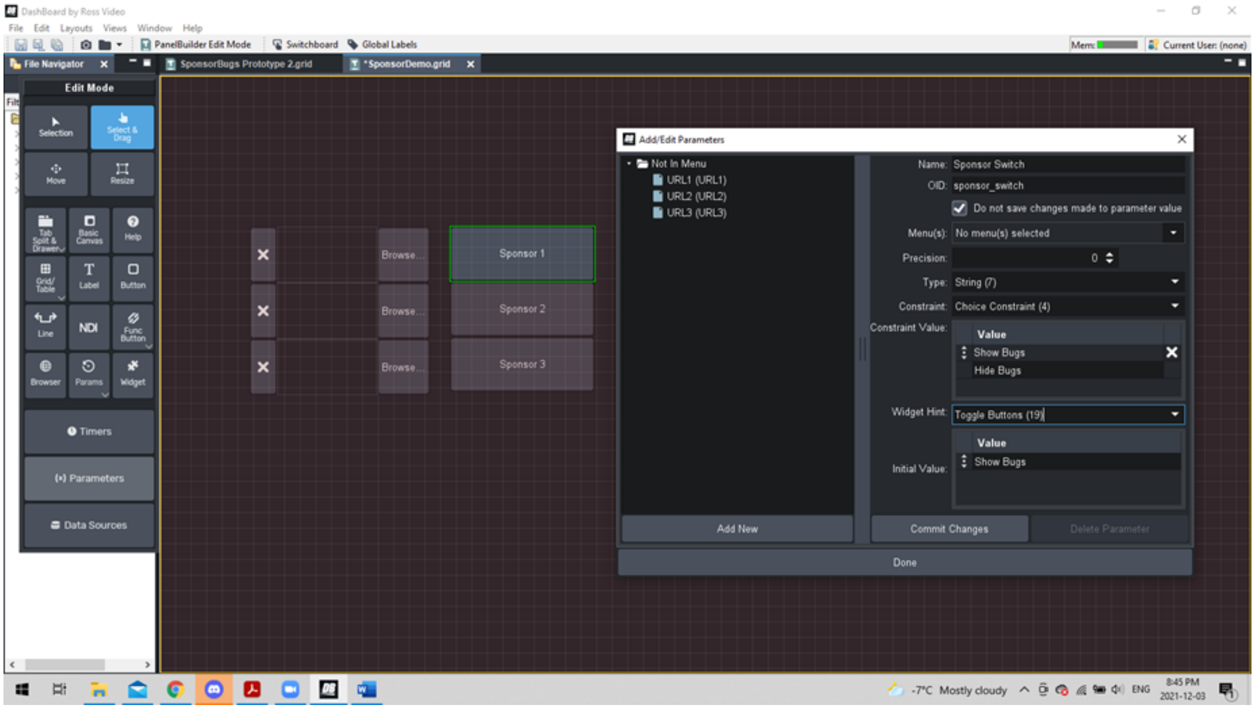
Click “Add” at the bottom, which will bring you to the following screen.



On the dropdowns on the right of the screen, click ogScript>Panel Elements>Reveal. When the blue Reveal box appears on the screen, click the id dropdown and select Tab 1. Repeat these steps for sponsor buttons 2 and 3, setting the Reveal id to Tab 2 and Tab 3, respectively.

Step 4. Creating a Hide/Show Sponsors Toggle Button

Create a new parameter with name “Sponsor Switch” and OID “sponsor_switch”. Check the box “Do not save changes to parameter value.” Set variable type to “String” and constraint to “Choice Constait”. Set widget hint to “Toggle Buttons”. Create 2 constraint values, “Show Bugs” and “Hide Bugs”. For initial value, type “Show Bugs”. Click Done.



Add/Edit Parameters

Name: Sponsor Switch

OID: sponsor_switch

☑ Do not save changes made to parameter value

Menu(s): No menu(s) selected

Precision: 0

Type: String (7)

Constraint: Choice Constraint (4)

Constraint Value:

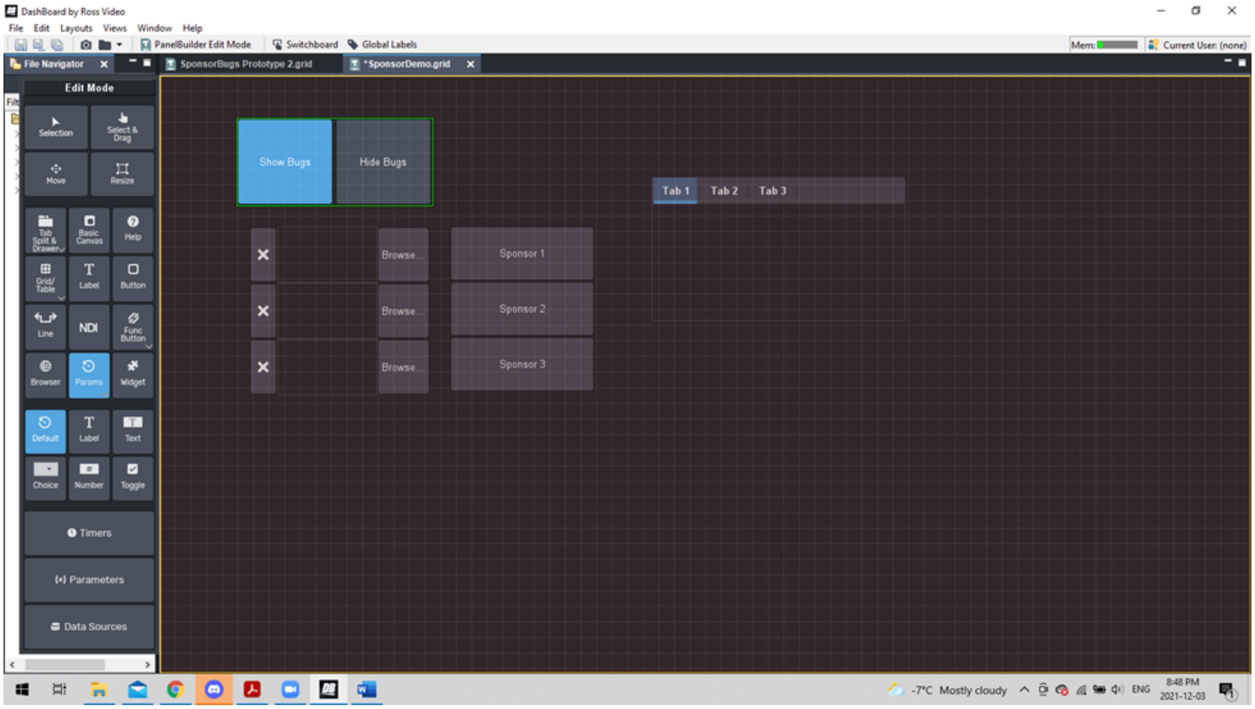
Value
Show Bugs
Hide Bugs

Widget Hint: Toggle Buttons (19)

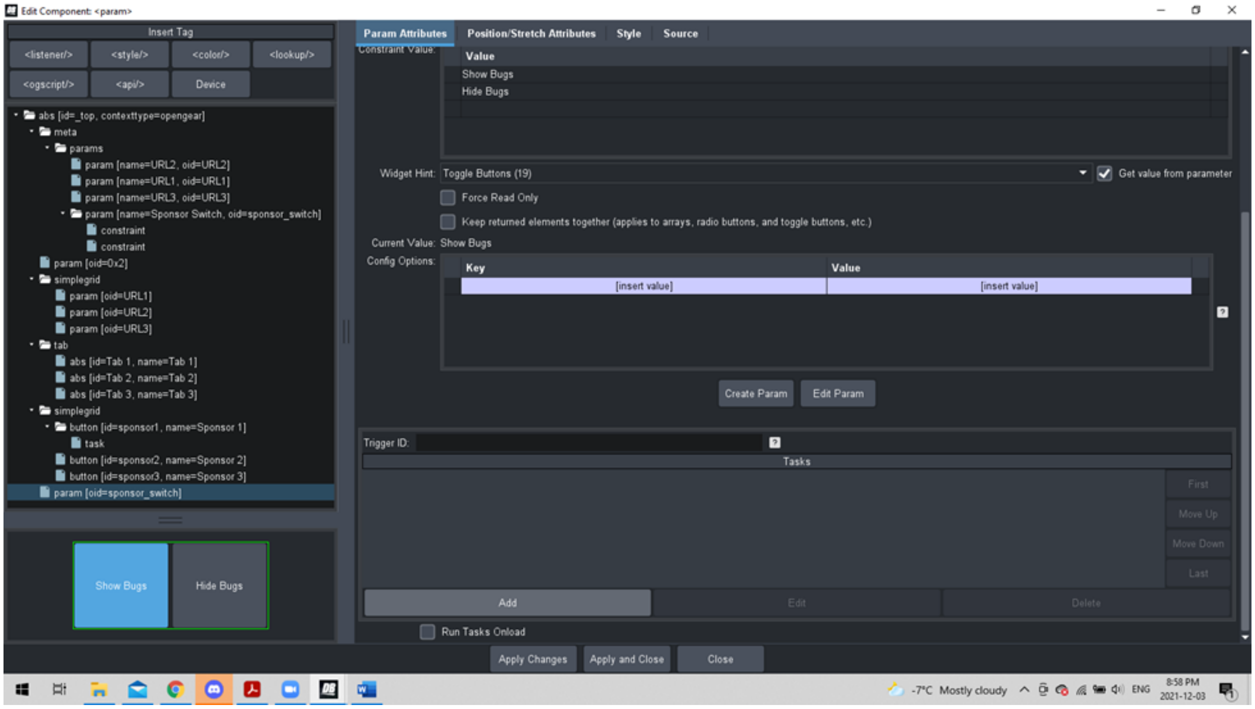
Initial Value:

Value
Show Bugs

In the toolbar, click “Params” and then “Default.” Now drag and drop where you would like the toggle button to be on the panel. When the window pops up asking to select a parameter, click “sponsor_switch” and then click OK.



Now double click the toggle button so we can give it an action. When the screen below appears, scroll to the bottom and click on the “Add” button.



- abs [id=_top, contexttype=opengear]
  - meta
    - params
      - param [name=URL2, oid=URL2]
      - param [name=URL1, oid=URL1]
      - param [name=URL3, oid=URL3]
      - param [name=Sponsor Switch, oid=sponsor_switch]
        - constraint
        - constraint
  - param [oid=0x2]
  - simplegrid
    - param [oid=URL1]
    - param [oid=URL2]
    - param [oid=URL3]
  - tab
    - abs [id=Tab 1, name=Tab 1]
    - abs [id=Tab 2, name=Tab 2]
    - abs [id=Tab 3, name=Tab 3]
  - simplegrid
    - button [id=sponsor1, name=Sponsor 1]
      - task
    - button [id=sponsor2, name=Sponsor 2]
    - button [id=sponsor3, name=Sponsor 3]
  - param [oid=sponsor_switch]

Value
Show Bugs
Hide Bugs

Widget Hint: Toggle Buttons (19) ☑ Get value from parameter

☐ Force Read Only

☐ Keep returned elements together (applies to arrays, radio buttons, and toggle buttons, etc.)

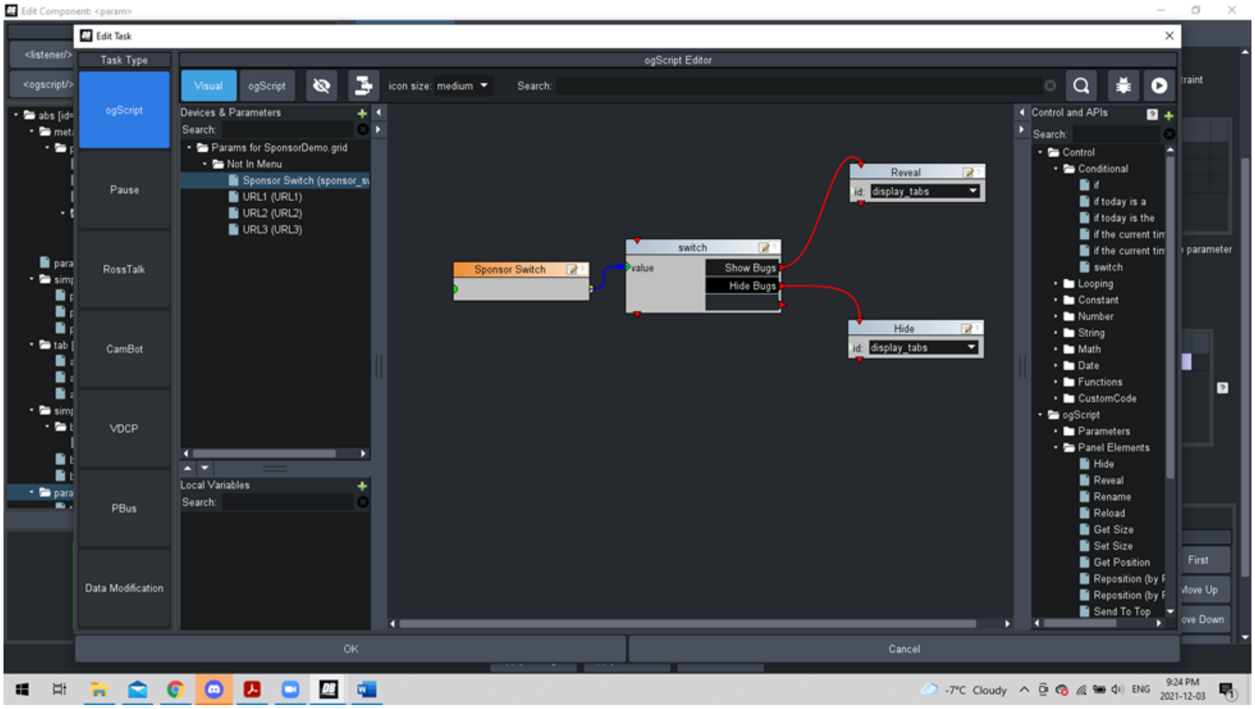
Current Value: Show Bugs

Config Options:

Key	Value
[insert value]	[insert value]

☐ Run Tasks Onload

Once you click add, you can configure the actions. Start by clicking the dropdown on the left, Params for *PanelTitle*.grid > Not in Menu > Sponsor Switch. When you do this, you will see the sponsor switch block appear on the visual editor. Now on the right, click Control > Conditional >Switch. Then click Panel Elements > Hide, and Reveal. Move the blocks around and connect them as shown below. Type the values “Show Bugs” and “Hide Bugs” into the switch block, and connect them to the reveal and hide blocks as shown below, which should be set to display_tabs. Click on OK when this is done.



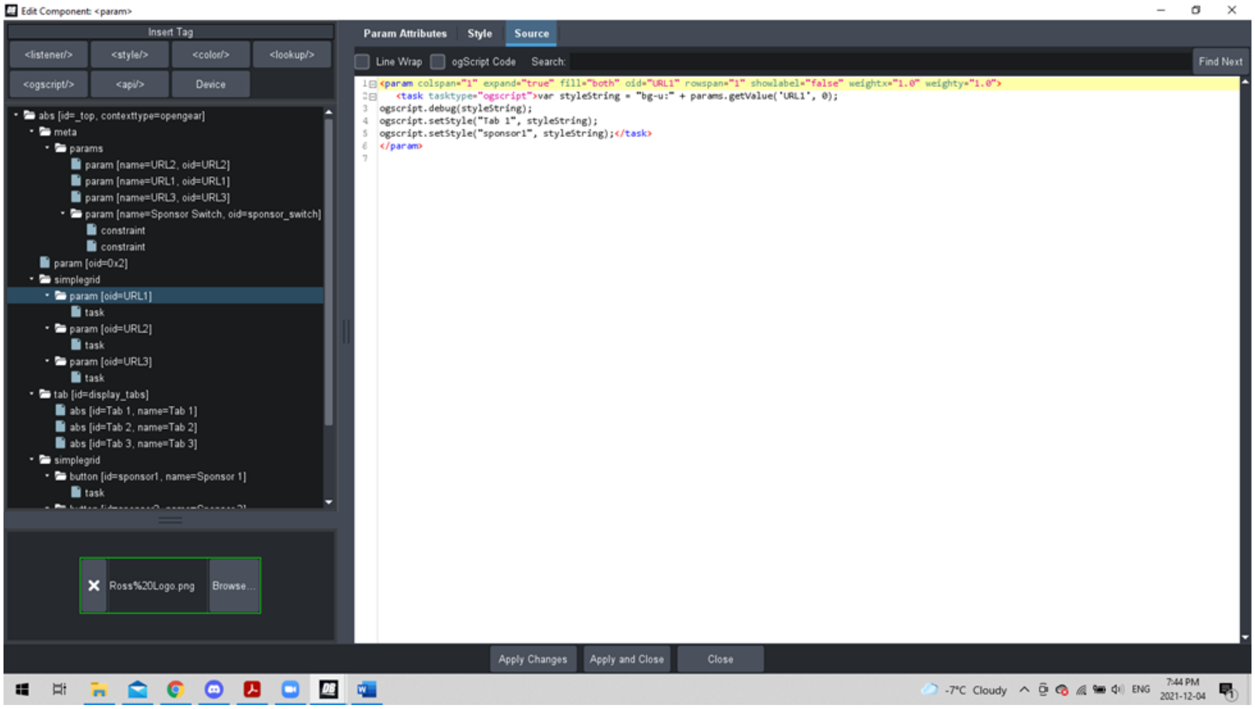
Step 5. Displaying the Photos

We want to display the photos in the output tabs, as well as in the buttons so that the user can easily see which sponsor they are going to display.

Using the select and drag tool, double click the first file picker. At the top of the page, click “Source”. Delete the existing code and replace it with the following:

```
<param colspan="1" expand="true" fill="both" oid="URL1" rowspan="1" showlabel="false" weightx="1.0" weighty="1.0">

  <task tasktype="ogscript">var styleString = "bg-u:" + params.getValue('URL1', 0);

ogscript.debug(styleString);

ogscript.setStyle("Tab 1", styleString);

ogscript.setStyle("sponsor1", styleString);</task>
```

</param>



```
<param colspan="1" expand="true" fill="both" oid="URL1" rowspan="1" showlabel="false" weightx="1.0" weighty="1.0">
   <task tasktype="ogscript">var styleString = "bg-u:" + params.getValue('URL1', 0);
ogscript.debug(styleString);
ogscript.setStyle("Tab 1", styleString);
ogscript.setStyle("sponsor1", styleString);</task>
</param>
```

Click on Apply and Close at the bottom of the page. Repeat these steps for the other 2 file pickers, replacing the numbers in the highlighted parts of the code with the respective number for each of the file pickers.

The panel is now completed and fully functional.

To Build Standings Panel

Standings User Manual

Step #1: Create a new dashboard panel by selecting “File”>New>New Custom Panel File. Name this panel “eastStandings”.

Step #2: Go to “PanelBuilder Edit Mode”

Step#3: Add a label that says “Eastern Conference” towards the middle of the screen.

Step #4: Double click on the label, then go Style>Font Size>Big

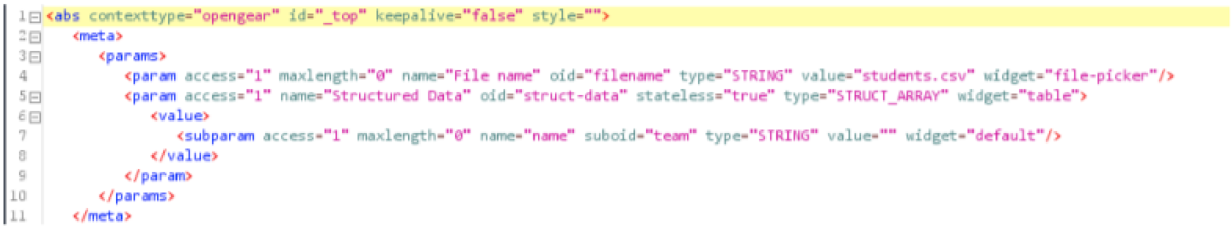
Step #5: Open the parameters menu

Step #6: On the right, create following:

Step #7:When you’re done, click “Commit Changes”, then “Done”. Next, click the “Select & Drag” tool, and double click on the background of the panel. A window similar to the image below will open, select the source tab at the top.

Step #8: Copy and paste the following code into your source:

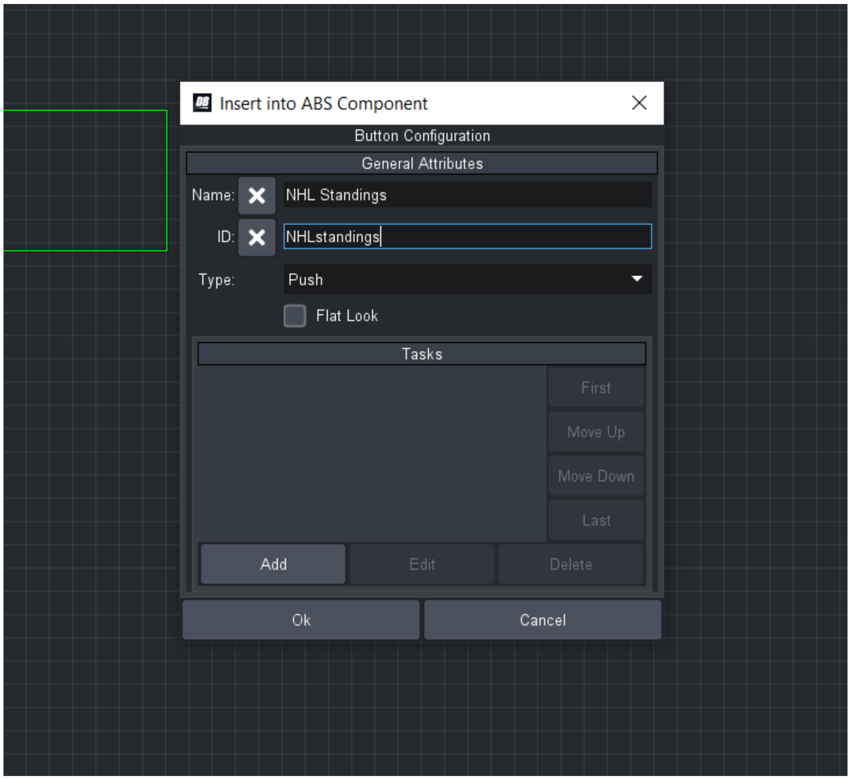
```
<param access="1" name="Structured Data" oid="struct-data"
stateless="true" type="STRUCT_ARRAY" widget="table">
<value>
<subparam access="1" maxlength="0" name="name"
suboid="team" type="STRING" value="" widget="default"/>
</value>
</param>
```



```
<abs contexttype="opengear" id="_top" keepalive="false" style="">
    <meta>
        <params>
            <param access="1" maxlength="0" name="File name" oid="filename" type="STRING" value="students.csv" widget="file-picker"/>
            <param access="1" name="Structured Data" oid="struct-data" stateless="true" type="STRUCT_ARRAY" widget="table">
                <value>
                    <subparam access="1" maxlength="0" name="name" suboid="team" type="STRING" value="" widget="default"/>
                </value>
            </param>
        </params>
    </meta>
```

Step #9: When you’re done, click apply and close at the bottom of the window, and reopen the parameters window, you should see the following:

Step #10: Create a Button, and title it “Read NHL Standings”



Name: NHL Standings

ID: NHLstandings

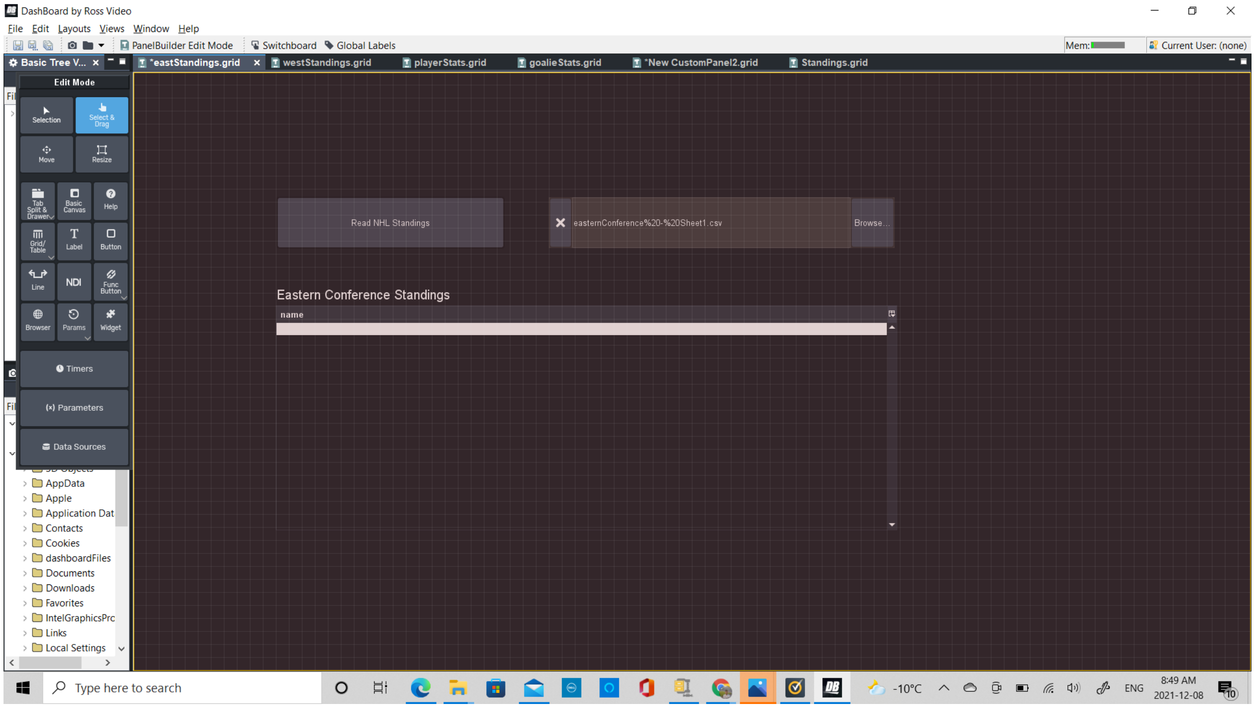
Type: Push

☐ Flat Look

Step #11: Click “Params” then “Default” and select an area just under the label “Eastern Conference Standings”, with a large length, and a considerable width as well. Then click on the “Structured Data” option and “Ok” when you’re done. It should look similar to the following.

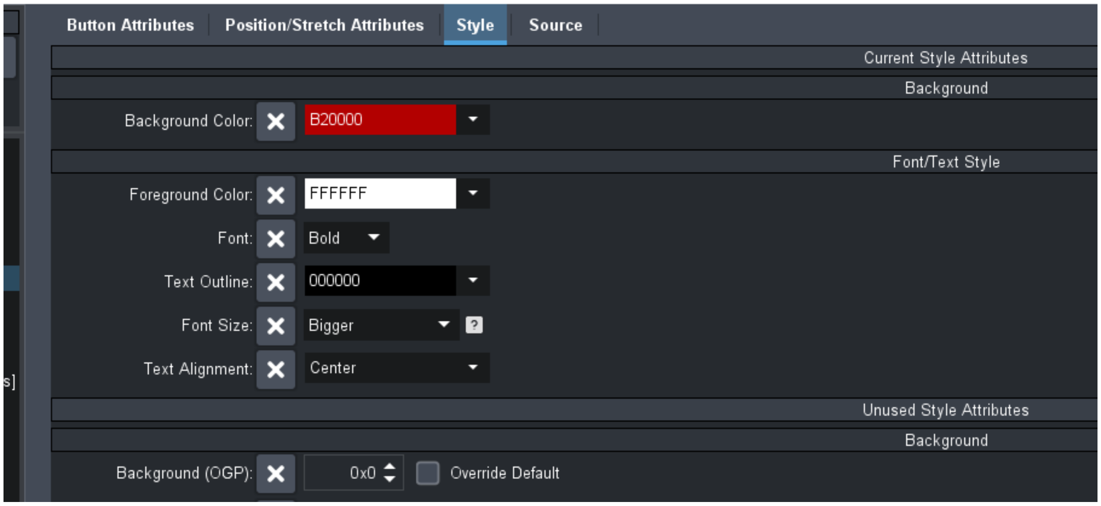
Step #12: Again, click on “Params” then “Default” and select an area on the right of the button. Then click on “File name” then click “Ok”.

Step #13: Your panel is now fully functional for the easternConference! Adding a csv file with the Eastern Conference standings will now display in your table! Let’s add some color to make it more visually appealing.



Step #14: Click on the “Select and Drag” tool, and double click on the background. Go to Style>Background Color>550109(Any Color of dark red works). Click “Apply and Close” when you’re done.

Step #15: Double click on the button. Go to Style>Background Color>B20000(Any Color of dark red works). Go to “Foreground Color:” and select “FFFFFF”(or white). Go to “Font:” and select Bold. Go to “Text Outline” and select 000000 (or black). Next go to “Font Size:” and select “Bigger”. Finally go to “Text Alignment:” and choose Center. Click “Apply and Close” when you’re done.



Background Color: B20000

Foreground Color: FFFFFF

Font: Bold

Text Outline: 000000

Font Size: Bigger

Text Alignment: Center

Background (OGP): 0x0 ☐ Override Default

Step #16: Double click on the structured data, and go to Style>Background Color>FFFFFF(white) and “Font Size” as Normal. Next select the “Even Rows” tab at the top.

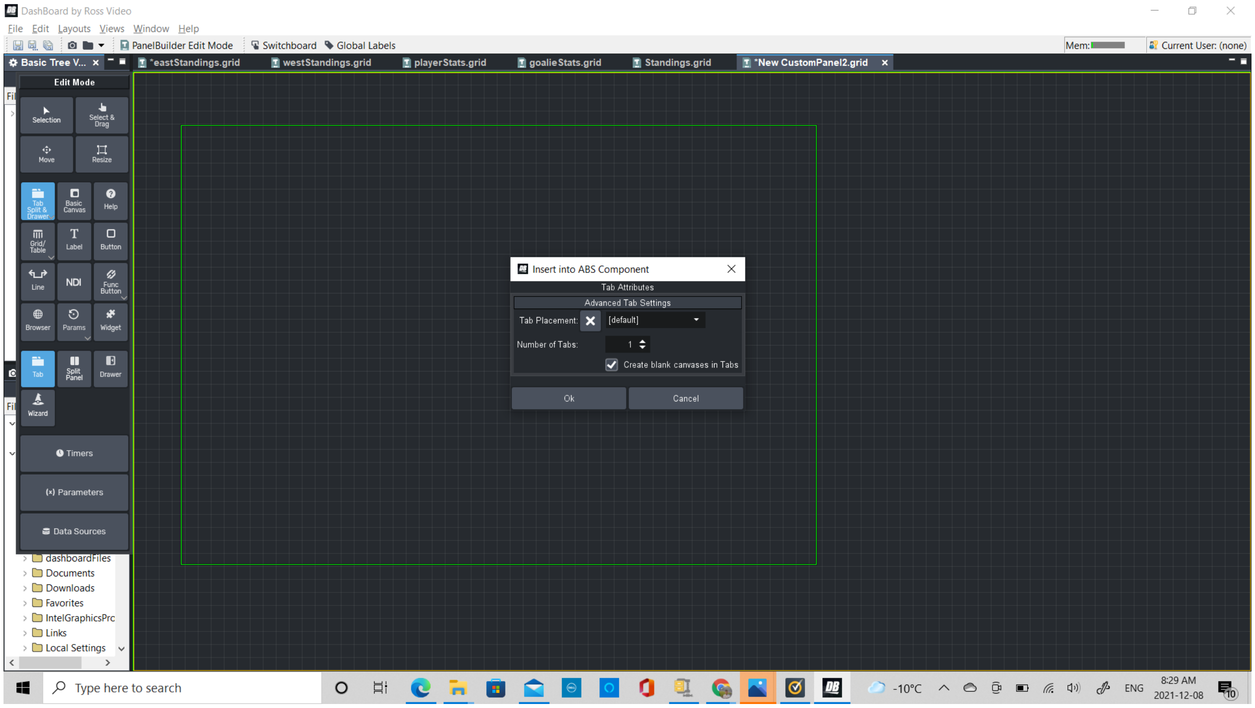
Change the “Background Color:” to D10808 or red. Change the “Foreground Color:” to FFFFFF or white. Next click the “Odd Rows” tab at the top, and change the “Background Color:” to FFFFFF or white, and the “Foreground Color:” to 000000 or black. Click “Apply and Close” when you’re done.

Step #17: Finally, double click on the file name widget, and go to Style>Background Color>474747(or a dark gray). Click Apply and Close when you’re done.

Step #18: Your panel for the eastern conference is now complete! Now repeat this process another 3 times: one for the western conference, one for the player standings, and one for the goalie standings. The process is the same thing, apart from a different label at the beginning.

Step #20: Create a new dashboard panel by selecting “File”>New>New Custom Panel File. Name this panel “Standings”.

Step #21: Go to “PanelBuilder Edit Mode” and click on the “Tab and Split Drawer” option, then the “Tab” option at the bottom. Select a very large area on the panel. Select 4 as the number of tabs.



Insert into ABS Component

Tab Attributes

Advanced Tab Settings

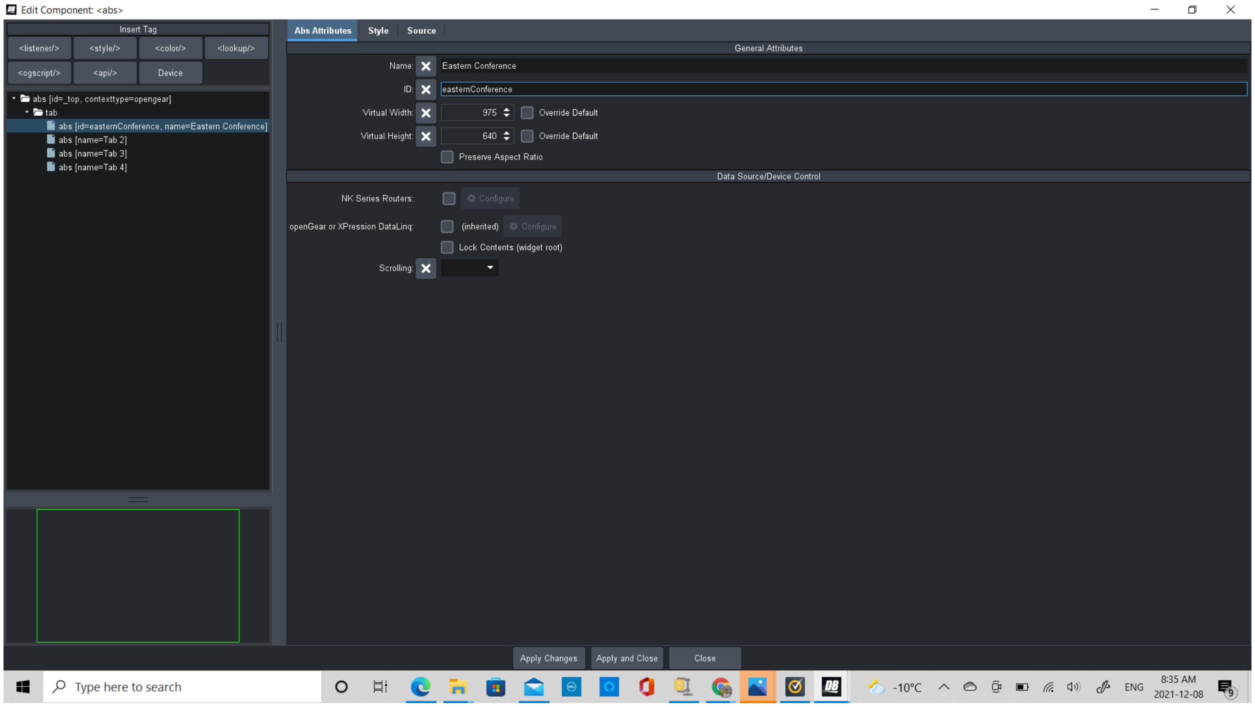
Tab Placement: [default]

Number of Tabs: 1

☑ Create blank canvases in Tabs

Ok Cancel

Step #22: Click the “Select and Drag” tool. Double click on the new Tab widget you just created, and change the name for all the tabs, to their respective titles, (Western Conference, Eastern Conference, etc.) Click Apply and Close when done.


Edit Component: <abs>

- abs [id=_top, contexttype=opengear]
  - tab
    - abs [id=easternConference, name=Eastern Conference]
    - abs [name=Tab 2]
    - abs [name=Tab 3]
    - abs [name=Tab 4]

General Attributes

Name: Eastern Conference

ID: easternConference

Virtual Width: 975 ☐ Override Default

Virtual Height: 640 ☐ Override Default

☐ Preserve Aspect Ratio

Data Source/Device Control

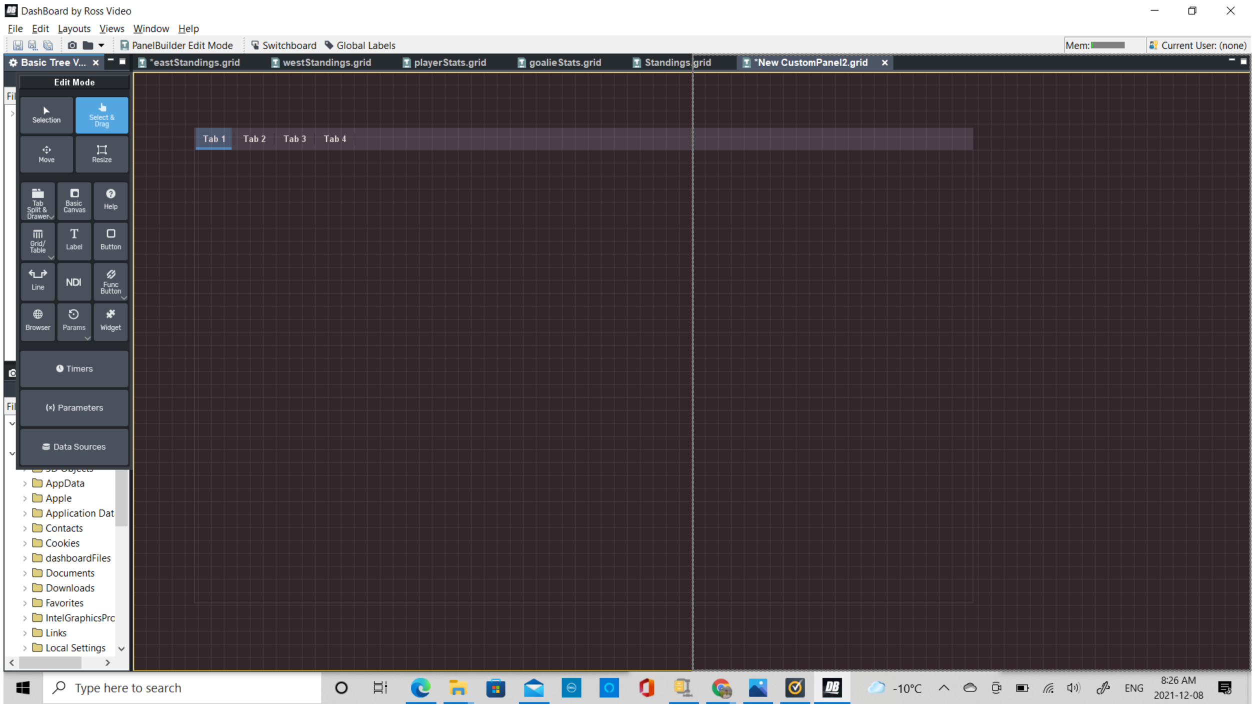
NK Series Routers: ☐

openGear or XPression DataLinq: ☐ (inherited)

☐ Lock Contents (widget root)

Scrolling:

Step #23: Select on your new “Standings” panel, and drag it to the far side of the screen until you see a little arrow pointing to the right. This will split your screen into two halves.



Step #23: Click in the middle of your easternConference panel, and drag it onto the “Standings” panel for the “Eastern Conference” tab. Then click on the “Western Conference” tab and drag and drop the westernConference panel. Repeat this process for all of your panels.

Step #24: Play around with the new “Standings” panel, until everything fits perfectly. Use the “Resize” tool to make the tab widget larger or smaller if necessary, and use the select and drag tool to move the panels within each tab, to make sure they’re all aligned.

Step #25: You’re done! Enjoy your fully functional, efficient, and attractive standings tab on dashboard.

To Build Fan Birthday Panel

Create a new panel in Dashboard by selecting “File”>New>New Custom Panel File. Name this panel “Fan Birthdays”.

Header

1. Insert a new table by going into “PanelBuilder Edit Mode”>Grid/Table>Table and drawing the table where you would like it on your grid.

2. Insert new Label. Name this label “Fan Birthdays”, or what you want to appear on the label. Style to preference under “style tab”. Import new image (This is for looks, skip if you’d like. If you want to style your button further and match it to a theme, you can create a new style tag and to match this label to your other buttons). This will be the Header for your panel.

Buttons Tab

1. Insert a new tab to store your information. Insert a new tab by going into “PanelBuilder Edit Mode”>Tab Split & Drawer>Tab and drawing the tabs where you would like them on your grid. Choose which configuration from the three provided. If you want a birthday button tab and a configuration tab, select 2 tabs when building. If you want a birthday button tab, a configuration tab, and a data table, select 3 tabs when building.

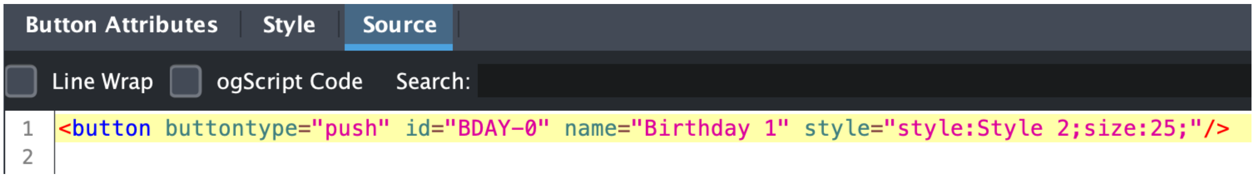
2. Insert a help button by going into “PanelBuilder Edit Mode”>Help and drawing the button where you would like it on your grid. You can fill this with any information you think will be helpful once you are out of “Panel Builder Edit Mode”.

3. Insert “Grid Table>Simple Grid with 4 columns”. Fill it with buttons and style to your liking (you can use the style tag that you created before - the advantage of a style tag is that if you decide you want to change something, you don’t have to go and change the style on everything. You can make one edit and everything automatically changes).

4. Rename each button to your liking. In this panel, we chose “Birthday 1”, “Birthday 2”... Give them each a name and an ID (Name: “Birthday 1” ID:BDAY-0).

5. Your source should look something like this (your style tag will be different, or if you do not have a tag it was display different information).

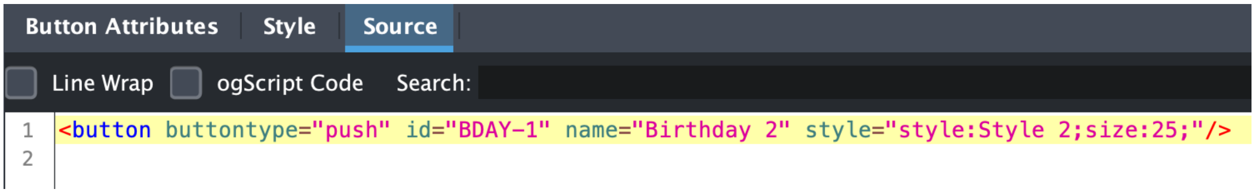
Ex. For first button



☐ Line Wrap ☐ ogScript Code Search:

```
<button buttontype="push" id="BDAY-0" name="Birthday 1" style="style:Style 2;size:25;"/>
```

Ex. For second button



☐ Line Wrap ☐ ogScript Code Search:

```
<button buttontype="push" id="BDAY-1" name="Birthday 2" style="style:Style 2;size:25;"/>
```

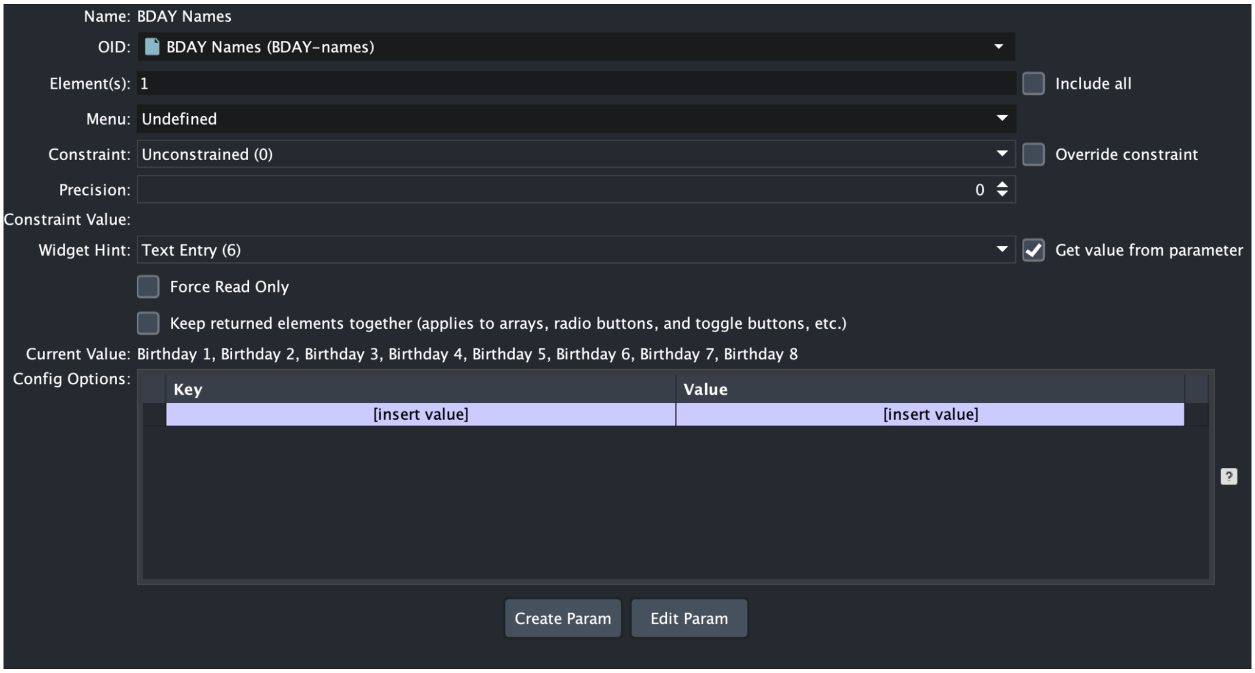
Buttons Config Tab

1. Insert “Grid Table>Table”. Set it up so that there are 8 columns and 10 rows (this is for creating a panel with 8 birthdays in mind- more can be added later on).

2. Add labels to the first and 6th rows. Adjust them so that each label covers two columns, with a rowspan of 2 (you should have 4 birthday labels in the 1st row, and 4 birthday labels in the 6th row). Name them “Birthday 1” to “Birthday 8”.

3. In your 2nd row, add a label called “Name” with a rowspan of 1.

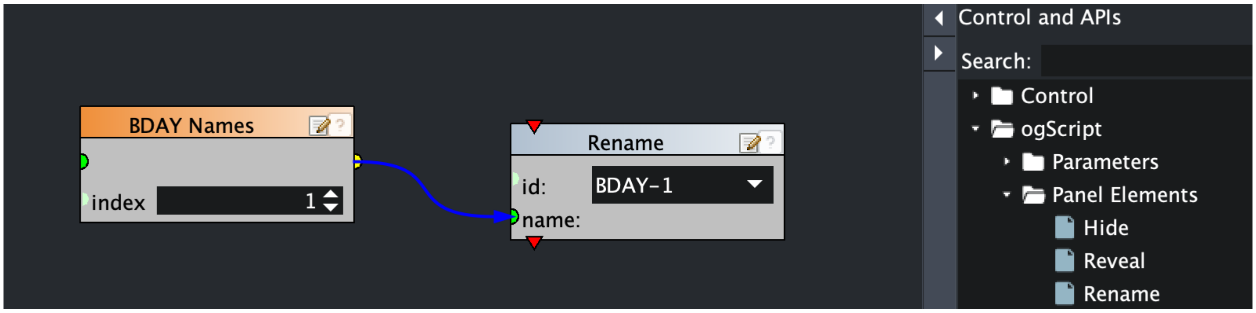
4. Set up the following Parameter to the right of the “Name” label. This will allow you to edit the name. Make the OID “BDAY Names”, and let it have the following. Make sure it is a text-entry parameter. To speed up the process, go into the source code and change the element number so that each birthday has a different element number:



Name: BDAY Names
OID: BDAY Names (BDAY-names)
Element(s): 1 ☐ Include all
Menu: Undefined
Constraint: Unconstrained (0) ☐ Override constraint
Precision: 0
Constraint Value:
Widget Hint: Text Entry (6) ☑ Get value from parameter
☐ Force Read Only
☐ Keep returned elements together (applies to arrays, radio buttons, and toggle buttons, etc.)
Current Value: Birthday 1, Birthday 2, Birthday 3, Birthday 4, Birthday 5, Birthday 6, Birthday 7, Birthday 8
Config Options:

Key	Value
[insert value]	[insert value]

5. In the visual editor, add a task to the parameter we just set up. We want the Name button on the “Button Tab” to rename when we change the name for a birthday. In ogScript, you can select “Rename” and then the button you wish to rename when you change the text. Your task will look like this:

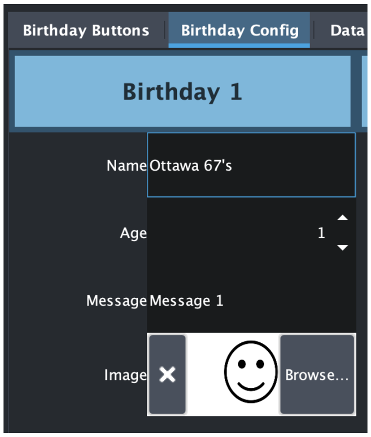


index 1

id: BDAY-1

name:

When you change the text in the Birthday Config Tab, the name of the person who is celebrating will appear in the Birthday Buttons Tab.



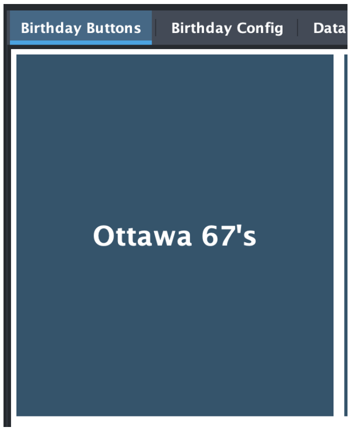
Name Ottawa 67's

Age 1

Message Message 1

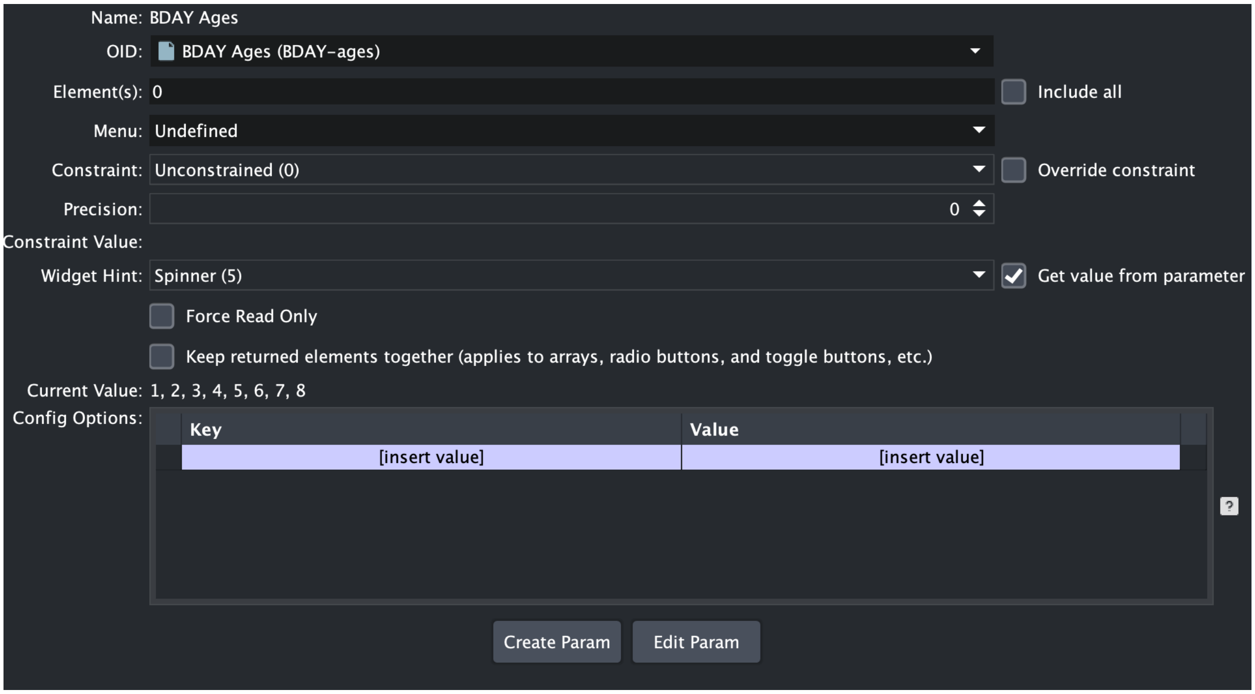
Image

→



6. In your 3rd row, add a label called “Age” with a rowspan of 1.

7. Set up the following Parameter to the right of the “Age” label. This will allow you to edit the age. Make the OID “BDAY Ages”, and let it have the following. Make sure it is a spinner. To speed up the process, go into the source code and change the element number so that each birthday has a different element number:



Name: BDAY Ages

OID: BDAY Ages (BDAY-ages)

Element(s): 0 ☐ Include all

Menu: Undefined

Constraint: Unconstrained (0) ☐ Override constraint

Precision: 0

Constraint Value:

Widget Hint: Spinner (5) ☑ Get value from parameter

☐ Force Read Only

☐ Keep returned elements together (applies to arrays, radio buttons, and toggle buttons, etc.)

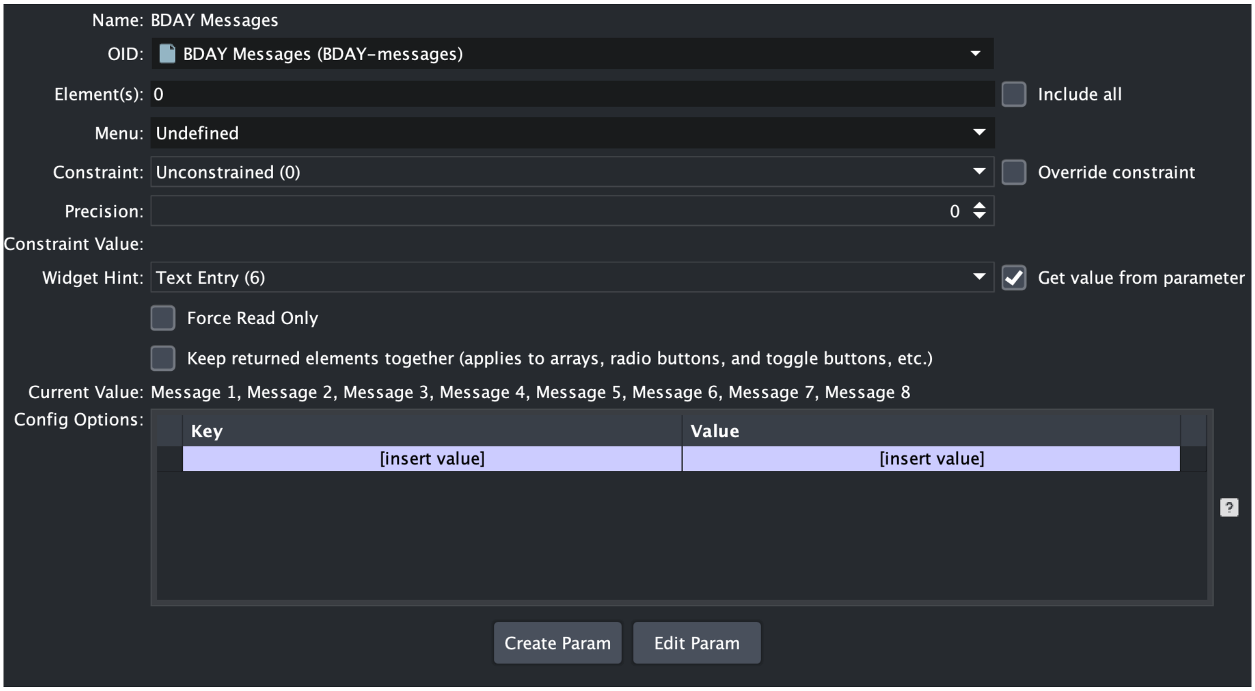
Current Value: 1, 2, 3, 4, 5, 6, 7, 8

Config Options:

Key	Value
[insert value]	[insert value]

8. In your 4th row, add a label called “Message” with a rowspan of 1. This is similar to the “BDAY Names” parameter we just set up.

9. Set up the following Parameter to the right of the “Message” label. This will allow you to edit the birthday message. Make the OID “BDAY Messages”, and let it have the following. Make sure it is a text entry. To speed up the process, go into the source code and change the element number so that each birthday has a different element number:



Name: BDAY Messages

OID: BDAY Messages (BDAY-messages)

Element(s): 0 ☐ Include all

Menu: Undefined

Constraint: Unconstrained (0) ☐ Override constraint

Precision: 0

Constraint Value:

Widget Hint: Text Entry (6) ☑ Get value from parameter

☐ Force Read Only

☐ Keep returned elements together (applies to arrays, radio buttons, and toggle buttons, etc.)

Current Value: Message 1, Message 2, Message 3, Message 4, Message 5, Message 6, Message 7, Message 8

Config Options:

Key	Value
[insert value]	[insert value]

10. Finally, we want to set up a place to browse for the fan images. In your 5th row, add a label called “Image” with a rowspan of 1. Add a browse option in the column to the right of the “Image” label.

11. You have now completed 1 full birthday button. Now that you have all the necessary features, it is easy to duplicate labels for other birthdays. Make sure that each button has its unique name.

12. If you find this Birthday Config tab too cluttered, you can add spacing between birthdays. Select “Birthday 5” in the config tab (should be the bottom left birthday), go to the source code, abs[name=Birthday Config]. Find part of source code that mentions 5,6,7,8. Copy that section of the code. Apply and close. Shift delete the row above to create spacing between each row.

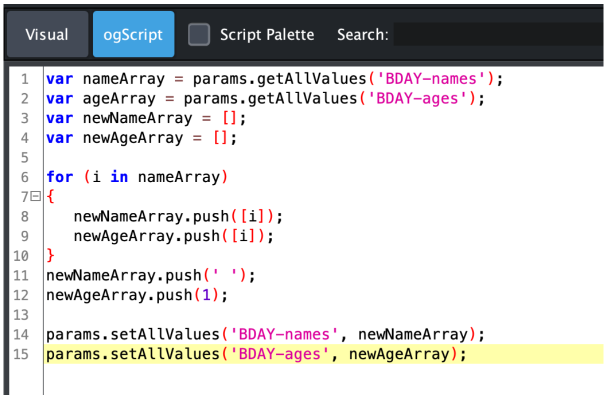
Data Table Tab

1. This tab is important because it contains everything that will be sent to XPression. Your birthday names, ages, messages, and images, are all constraints here.

2. Create a table with 4 columns and constraints as “BDAY Names”, “BDAY Ages”, “BDAY Messages”, and “BDAY Images”.

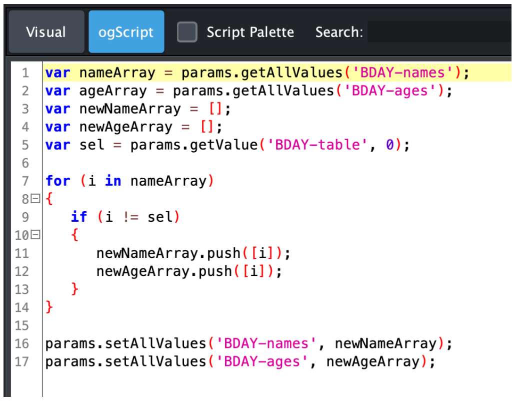
3. You can also add two buttons on the bottom to make tasks even easier to accomplish. Let’s call them “Add new Line” and “Delete Line”. Add a task to each button.

For “Add new Line”:



```
var nameArray = params.getAllValues('BDAY-names');
var ageArray = params.getAllValues('BDAY-ages');
var newNameArray = [];
var newAgeArray = [];

for (i in nameArray)
{
   newNameArray.push([i]);
   newAgeArray.push([i]);
}
newNameArray.push(' ');
newAgeArray.push(1);

params.setAllValues('BDAY-names', newNameArray);
params.setAllValues('BDAY-ages', newAgeArray);
```

For “Delete Line”



```
var nameArray = params.getAllValues('BDAY-names');
var ageArray = params.getAllValues('BDAY-ages');
var newNameArray = [];
var newAgeArray = [];
var sel = params.getValue('BDAY-table', 0);

for (i in nameArray)
{
   if (i != sel)
   {
      newNameArray.push([i]);
      newAgeArray.push([i]);
   }
}

params.setAllValues('BDAY-names', newNameArray);
params.setAllValues('BDAY-ages', newAgeArray);
```

4. The advantage of the data table tab is that any information you change also changes in the Config Tab automatically because everything is synced when you create parameters.

Timers, Display, In and Out Buttons

1. Insert a new Simple Grid to the right of the 3 tabs, and fill it with buttons. Make the dimensions for this 8 rows by 2 columns, which gives you enough to show 8 birthdays. This can always be modified later. You can label each button on the left “Birthday X In” for all birthdays, and the buttons on the right “Birthday X Out” for all birthdays.

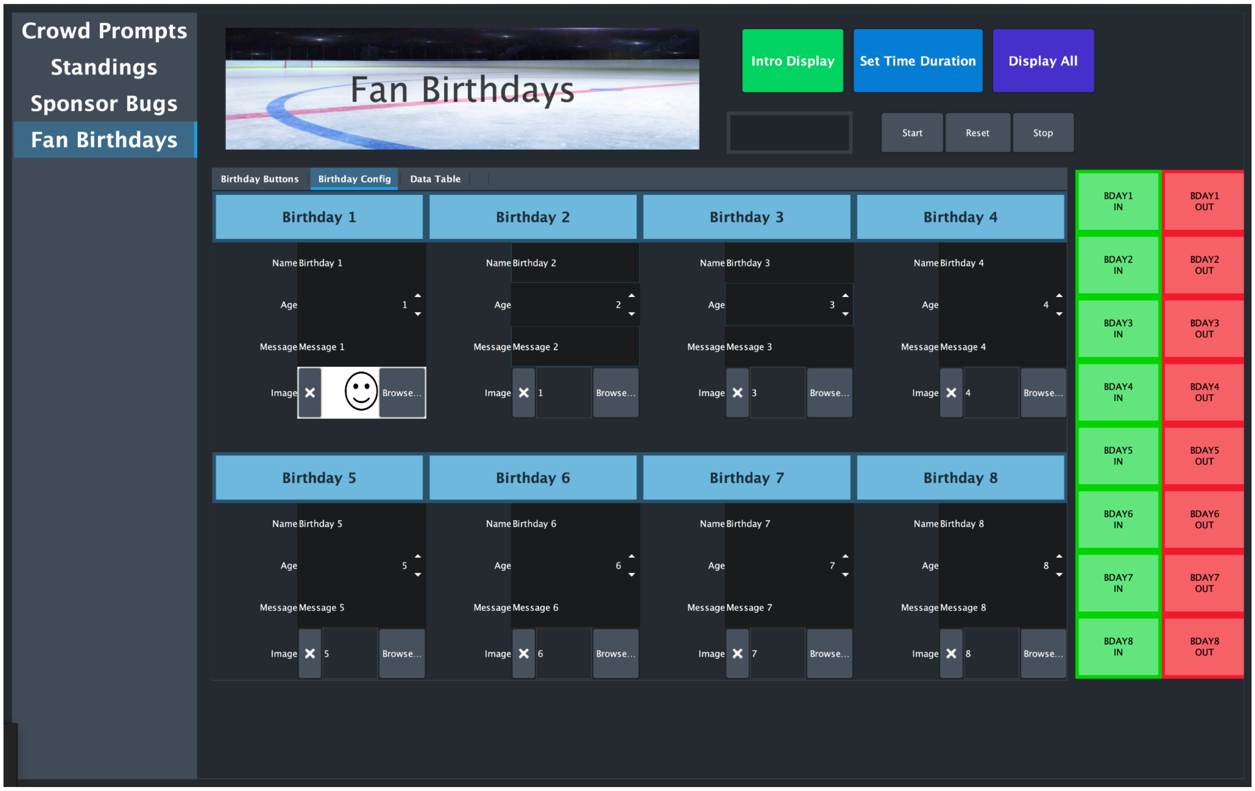
2. Create 2 style tags: one that is green for “In” and one that is red for “Out”. This makes the design easy to understand as green means go and red means stop. The operator will be able to easily tell which buttons perform which tasks. Apply your green style tag to all the birthday “In: buttons and the red style tag to all the birthday “Out” buttons.

3. Insert 3 buttons at the top of your panel. Name them “Intro Display”, “Set time duration”, and “Display all”. These will be the buttons that will display “Happy Birthday to…” (or the message you would like to display), modify the time that each birthday is shown for, and display all the birthdays. We will add tasks to these buttons later.

4. Add a timer as well as a start, pause, and reset button. Have it set up as a minute:second timer. This will be the timer that controls when birthdays are shown on screen.

5. Add a timertask that pushes the “In” buttons in order in a certain period of time. This is how the information will be automatically sent to XPression and displayed at a certain time interval. You can also override the timertask if necessary. The “In” and “Out” buttons are still fully functional and can be pressed if you only want to show one birthday or need to remove one quickly.

Your Final Panel should look something like this, once you have added style tags and modified colours:



Birthday 1
Name Birthday 1
Age 1
Message Message 1
Image

Birthday 2
Name Birthday 2
Age 2
Message Message 2
Image 1

Birthday 3
Name Birthday 3
Age 3
Message Message 3
Image 3

Birthday 4
Name Birthday 4
Age 4
Message Message 4
Image 4

Birthday 5
Name Birthday 5
Age 5
Message Message 5
Image 5

Birthday 6
Name Birthday 6
Age 6
Message Message 6
Image 6

Birthday 7
Name Birthday 7
Age 7
Message Message 7
Image 7

Birthday 8
Name Birthday 8
Age 8
Message Message 8
Image 8

To Integrate all Panels

1. Open all .grid files in Dashboard. Select and drag the one tab, for example the “Crowd Prompts” panel to the right side of Dashboard until a small white arrow appears that points to the right edge of your panel. Drop the file. You should have each panel side by side.

2. On the left side of the “Crowd Prompts” panel, create a new panel in Dashboard by selecting “File”>New>New Custom Panel File. Name this panel.

3. Select this panel, and open Panel Builder Edit Mode.

4. Insert a New Tab by going into “PanelBuilder Edit Mode”>Tab Split & Drawer>Tab and drawing the tabs where you would like them on your grid.

5. Add as many tabs as you have panels you wish to combine. Name these tabs.

6. Select this panel, and open Panel Builder Edit Mode. Select the “Crowd Prompts” panel, and open Panel Builder Edit Mode.

7. Click anywhere on the “Crowd Prompts” grid. Drag and drop the file into the new custom panel with the Tabs. Now, your “Crowd Prompt” panel is in the new custom panel.

8. Repeat these steps for each panel you have. Resizing may be necessary.

MakerRepo

All panel files and description of stadium fan experience can be found at this link:
https://makerepo.com/rjohn100/1036