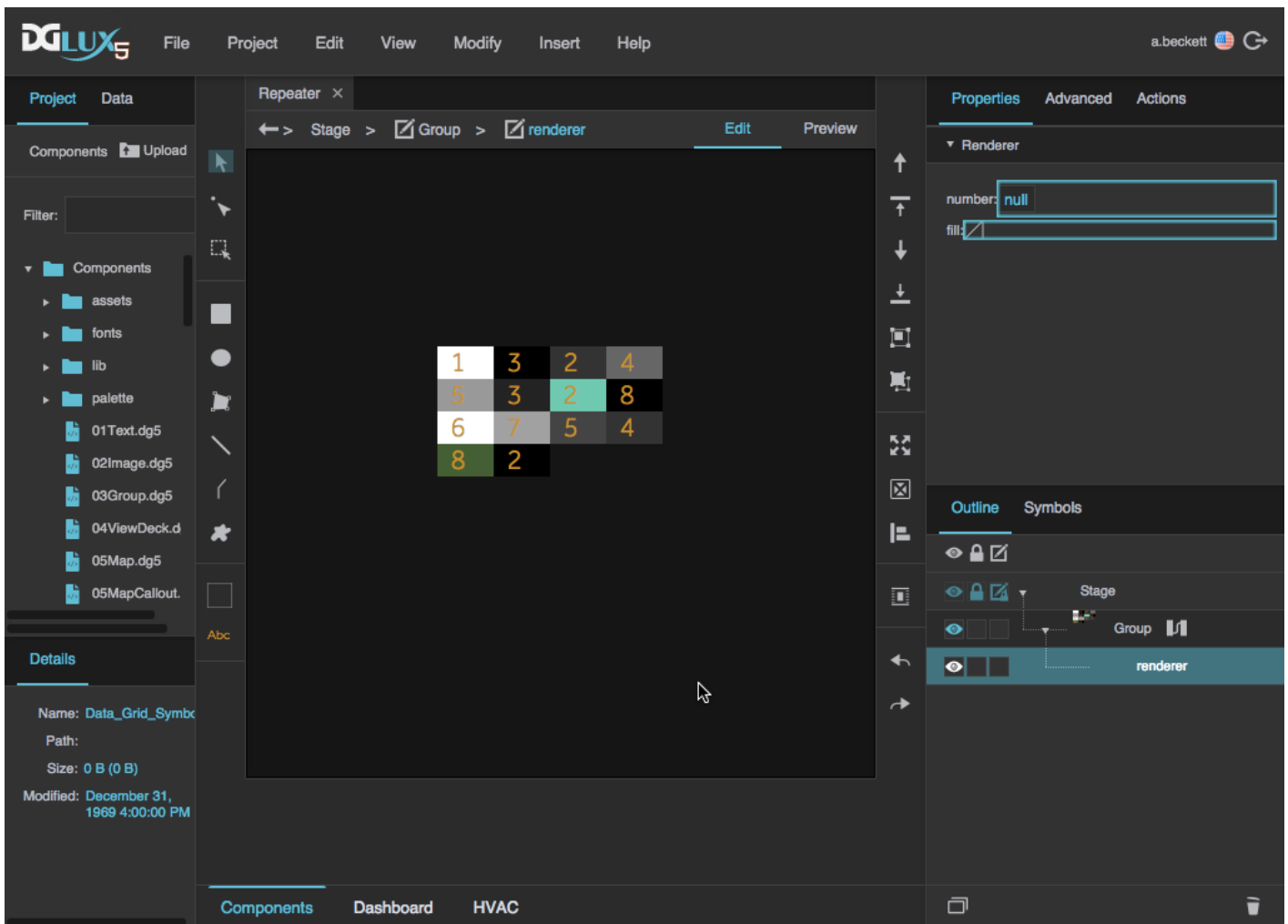


Repeaters

A repeater takes a [symbol](#) and renders an instance of that symbol for each row of data in a [table](#). Each instance of the symbol can have different parameter values, based on the data in the given row.

This page covers steps to make a simple repeater using the [group](#) component. You can also create repeaters for [view decks](#), [Google Maps](#), [map groups](#), and [data grids](#). A similar effect can be created for the markers, bars, or columns in a [chart](#) using an Item Renderer.

DGLux5 also features a [Raw SVG Repeater](#) that repeats an SVG instead of a symbol.



A repeater created using a group component in DGLux5

Video Tutorial: Repeaters

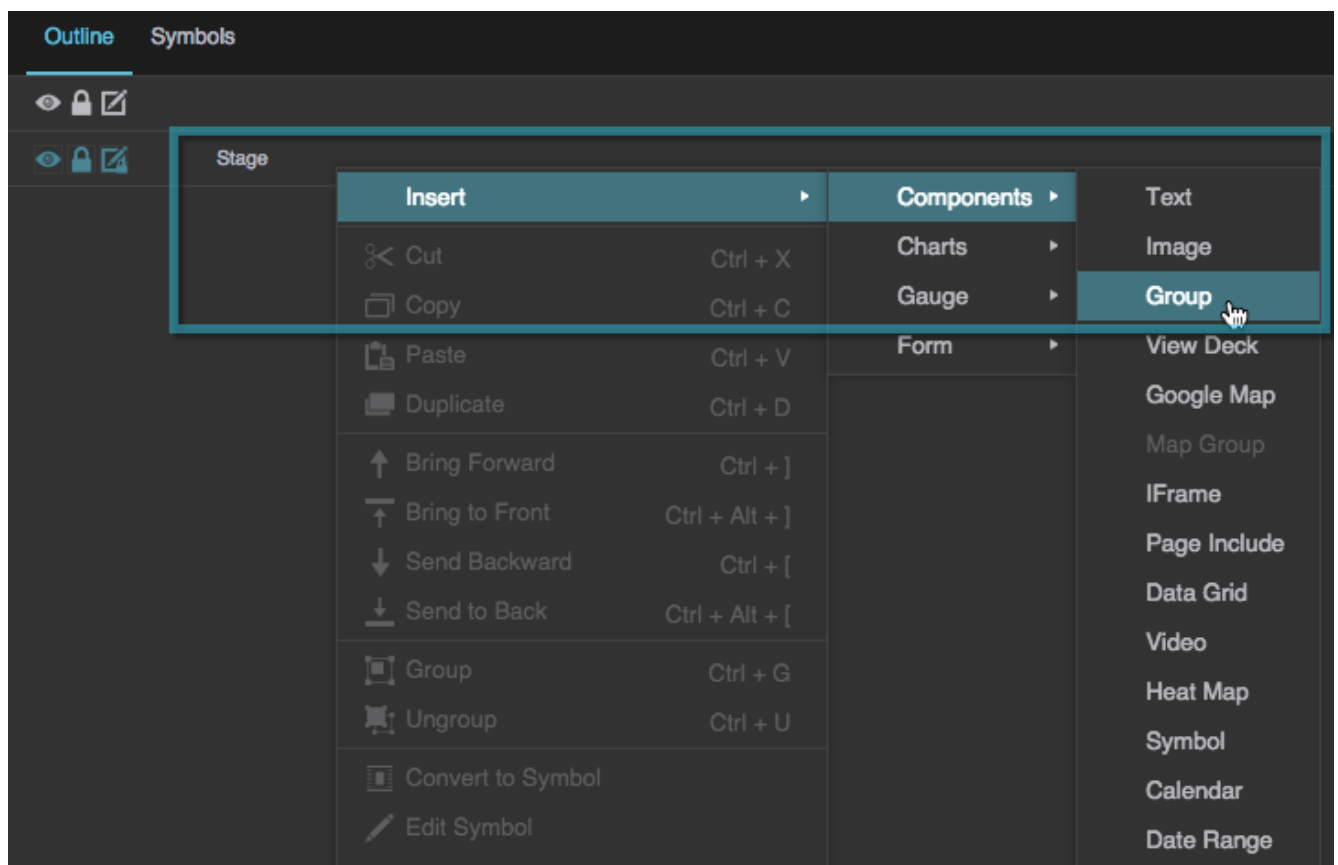
More video tutorials are [here](#).

Create a Simple Repeater

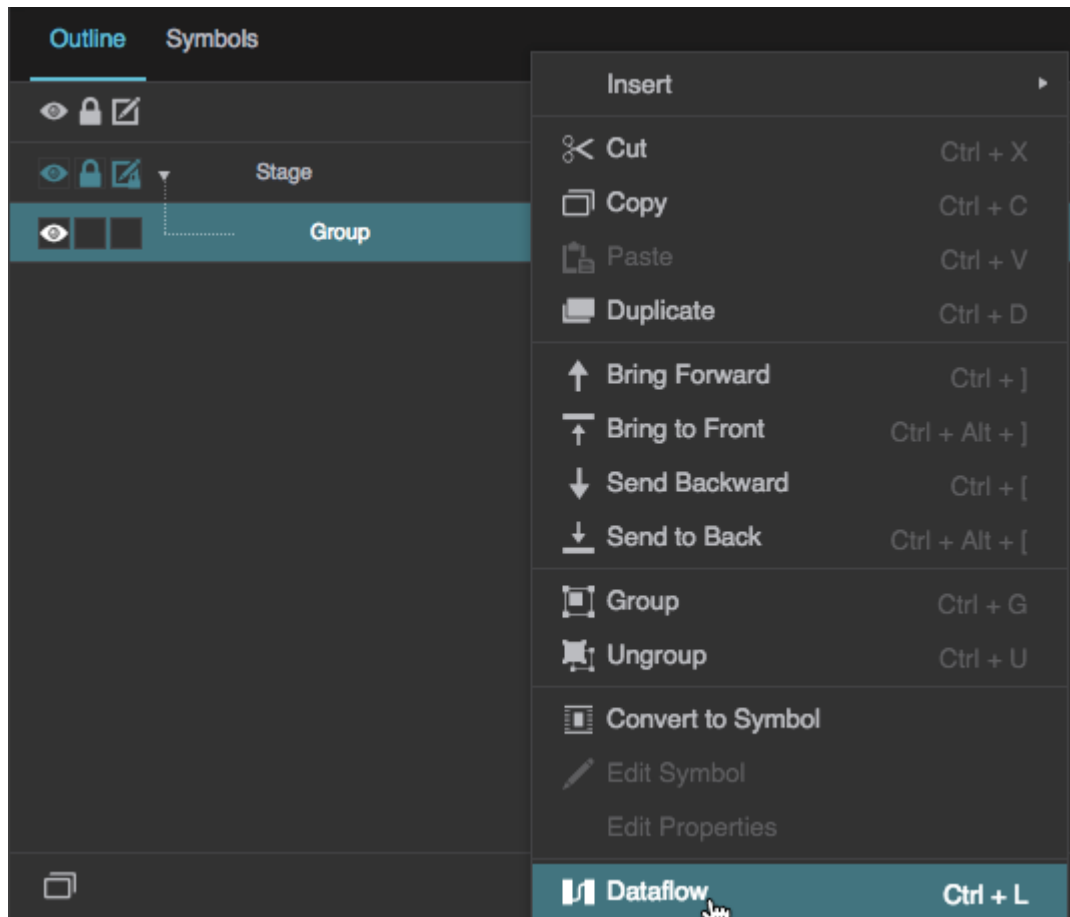
These steps show you how to create a simple repeater using a [group](#) component. In these steps, you will use a [Table](#) dataflow block and a [symbol](#).

Configure the Table

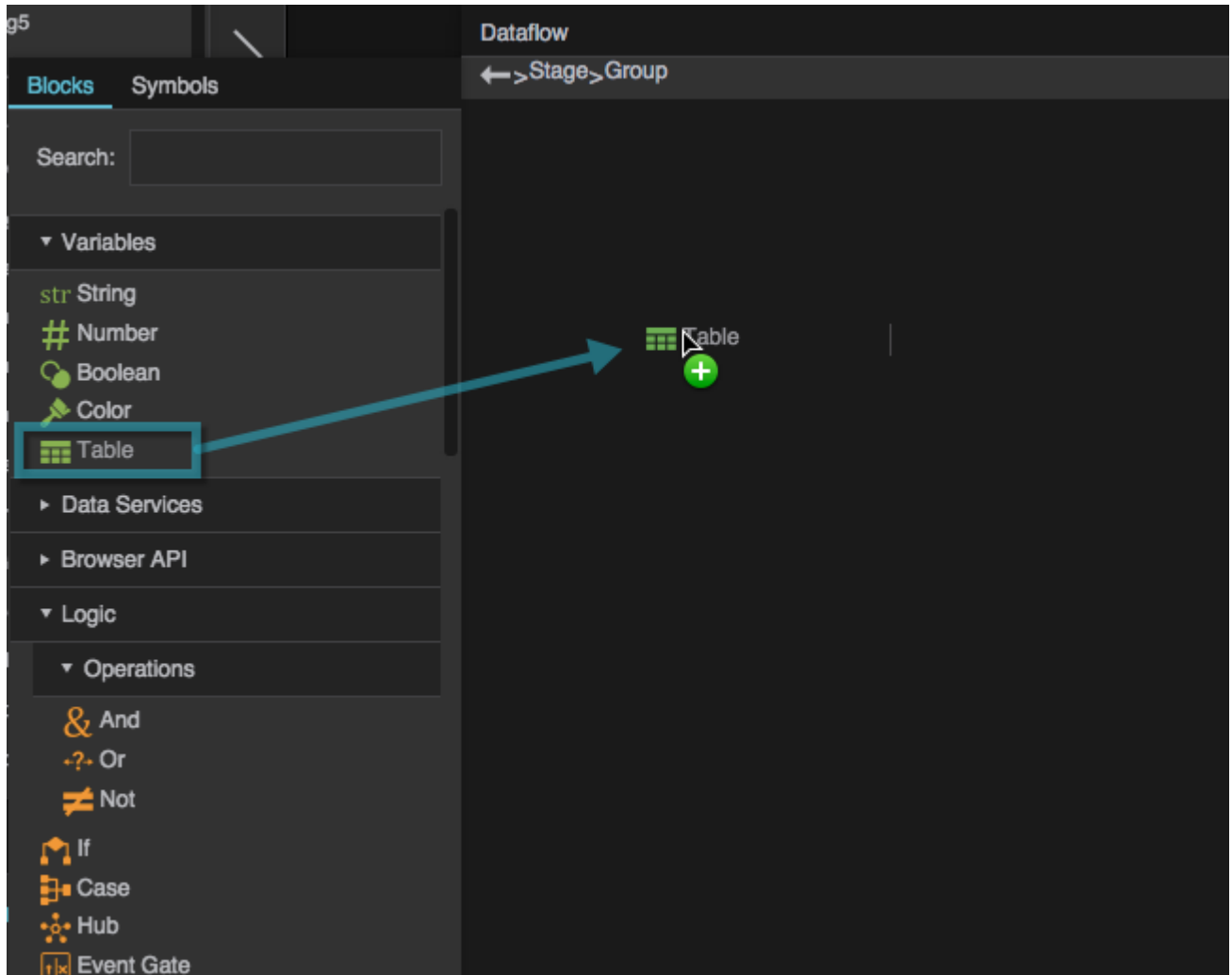
1. In the [Document window](#) or [Outline](#), right-click the Stage and select **Insert > Components > Group**.



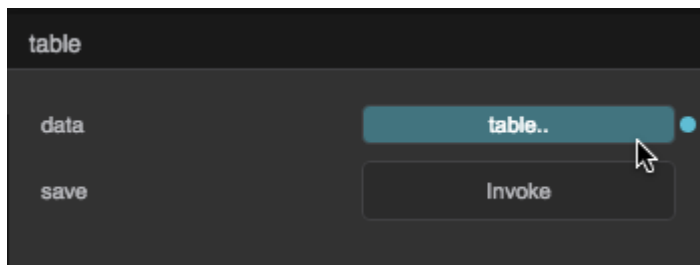
2. Right-click the group and select **Dataflow**.

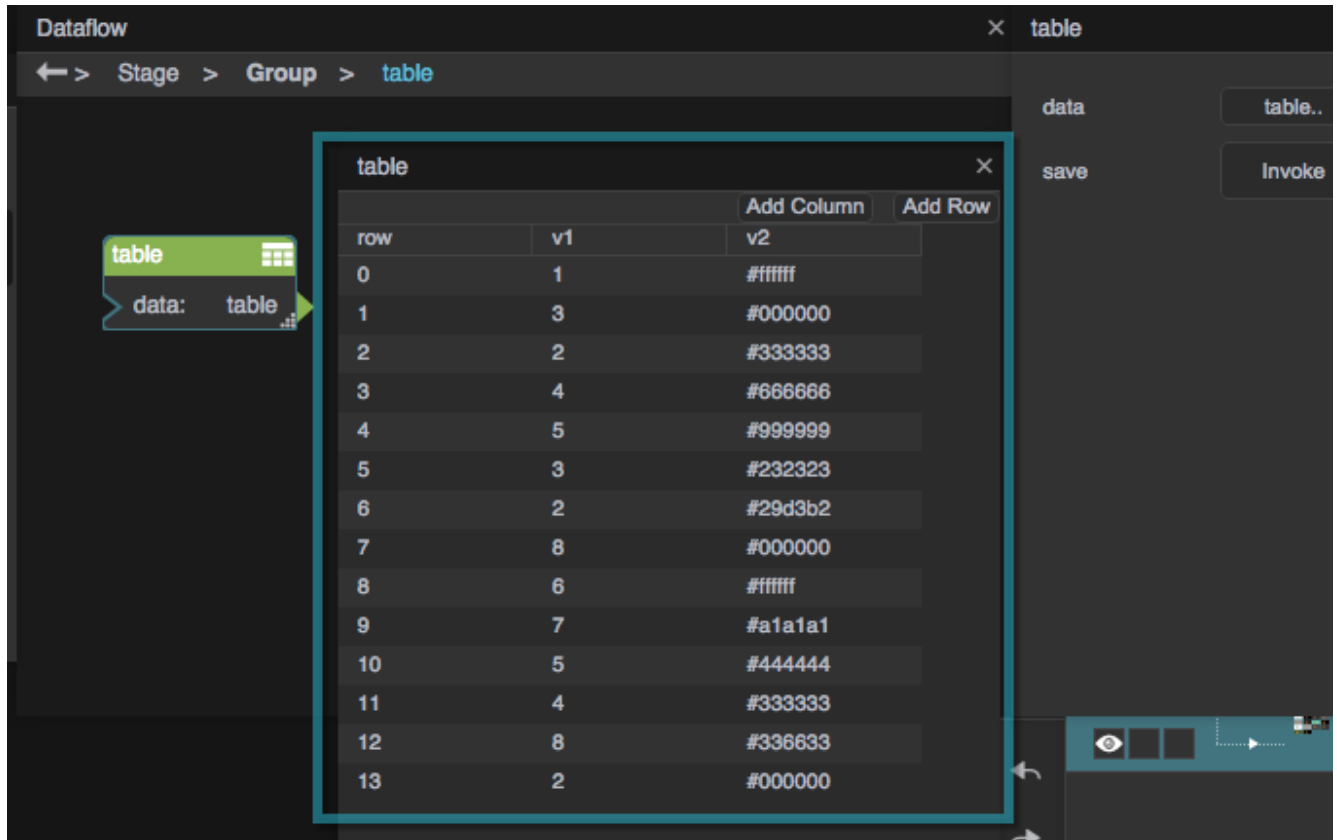


3. Drag a [Table](#) dataflow block to the dataflow window.



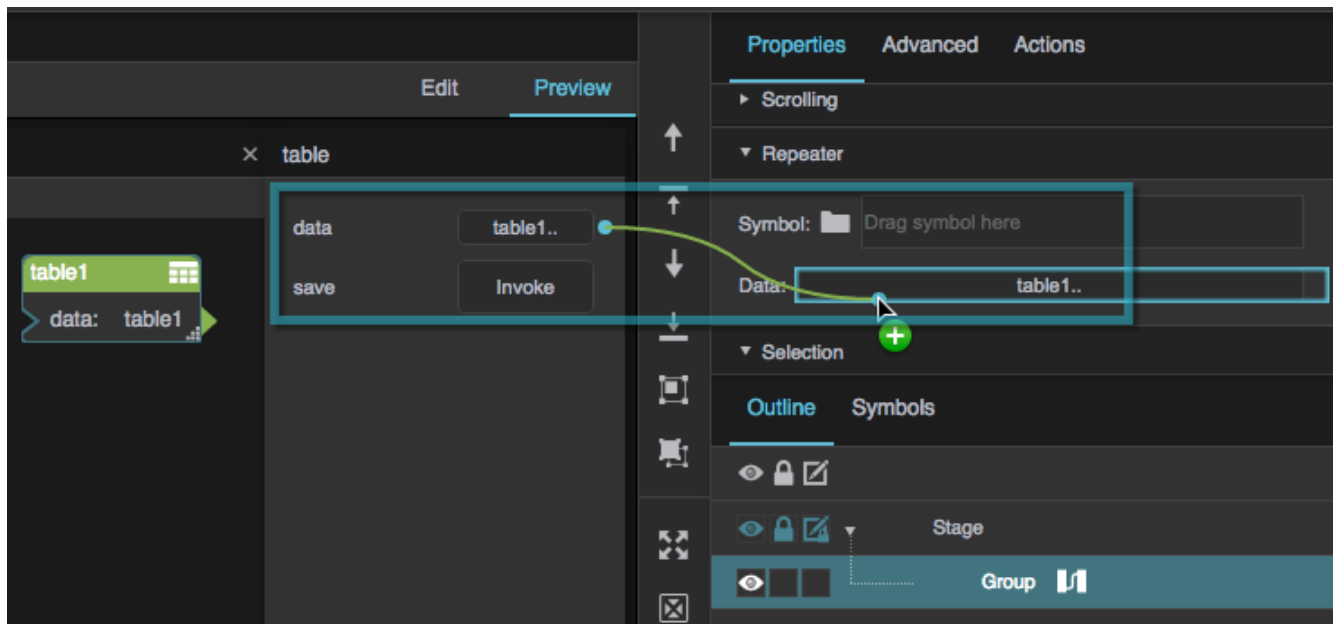
4. Click the button for the **data** property to open the table, and then edit the table so that it appears similar to the image below.






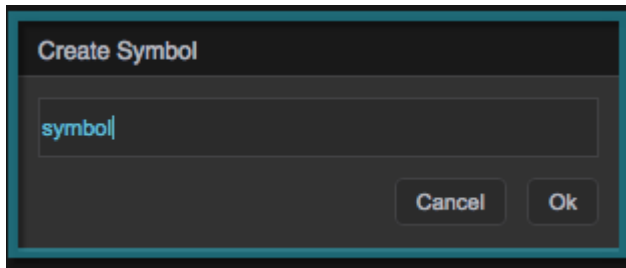
In the image, the first column contains integers, and the second column contains hexadecimal color codes. For help, see [Enter Data in a Table Dataflow Block](#).

- 5. Without closing the dataflow, select the group in the Outline.
- 6. In the [Property Inspector](#), under **Repeater**, bind the table for your repeater to the **Data** property.

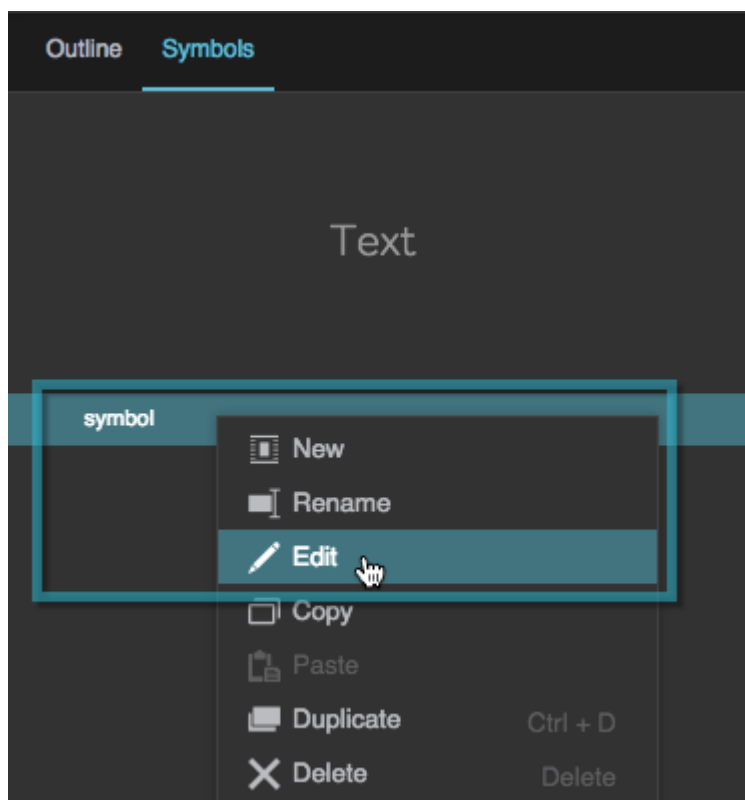


Create the Symbol

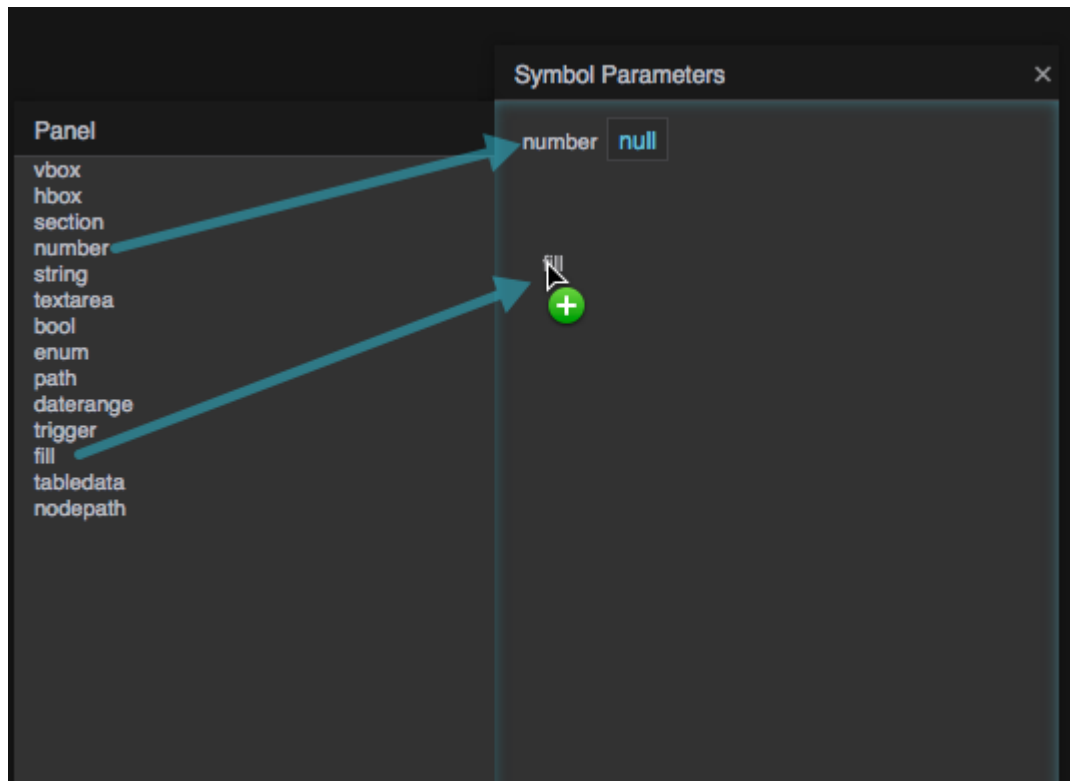
1. Create a [symbol](#) for your repeater:
 1. Right-click the Stage, and select **Insert > Components > Text**.
 2. Right-click the text component that you added, and select  **Convert to Symbol**.
 3. Name the symbol `symbol`, and click OK.



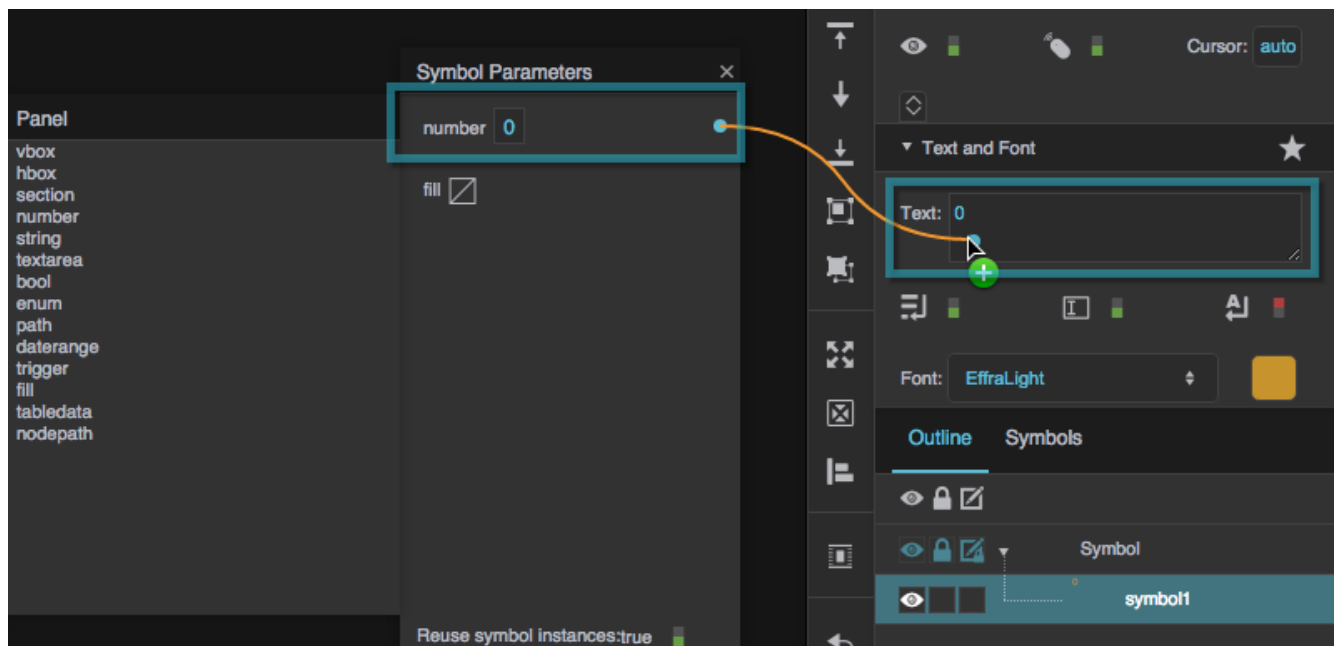
2. In the [Symbols panel](#), select **symbol**. Then, right-click **symbol**, and select **Edit**.



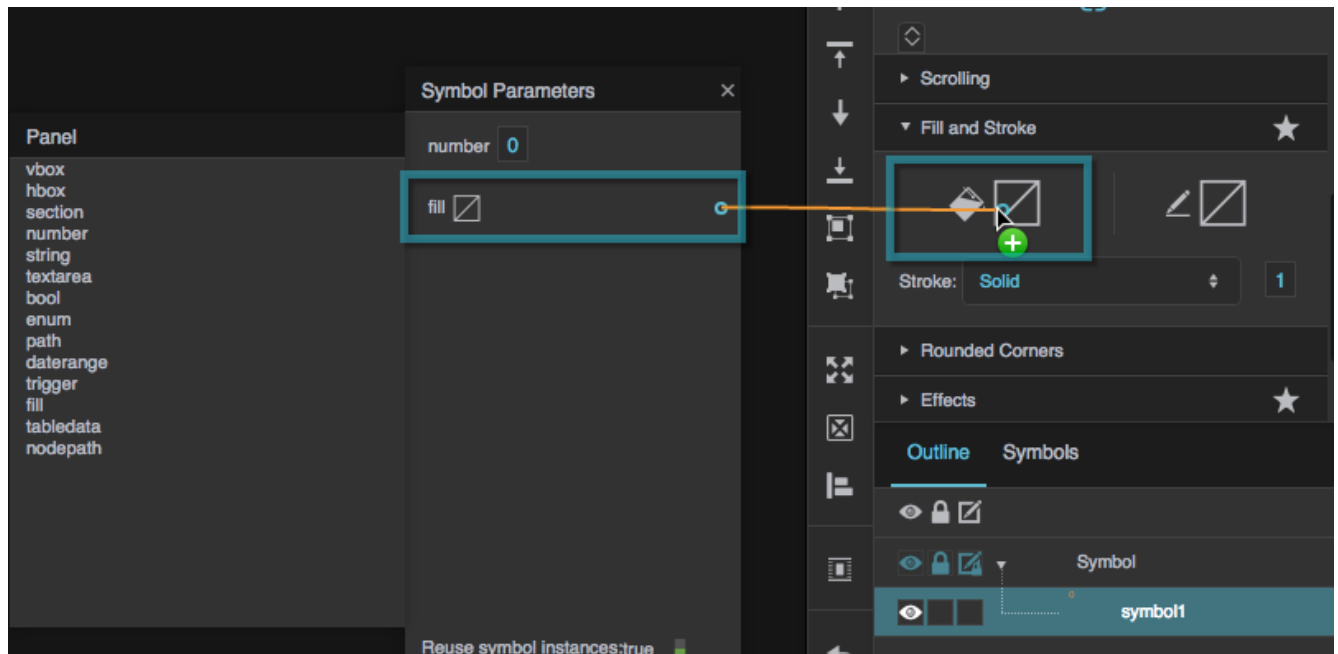
3. In the Outline, right-click **symbol**, and select **Edit Properties**.
4. In the [Symbol Parameters](#) dialog, drag one **number** and one **fill** to the right-hand panel.



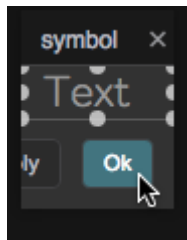
5. Bind the **number** parameter to the **Text** property in the Property Inspector.



6. Bind the **fill** parameter to the **Fill** property in the Property Inspector.



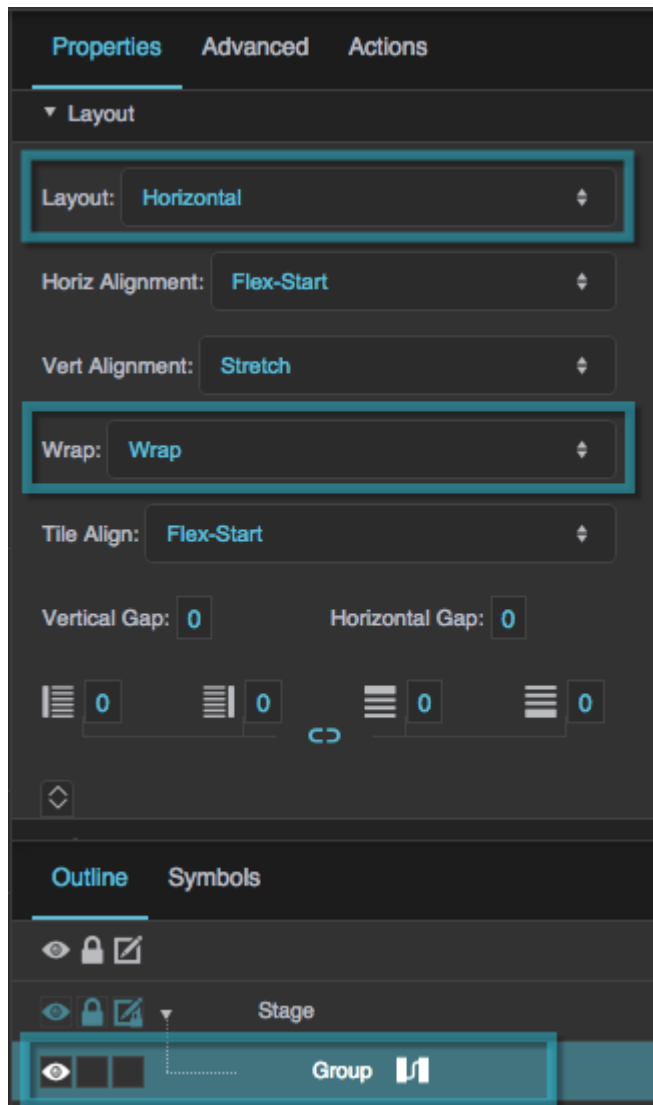
7. Close the Symbol Parameters dialog, and then click **OK** to close the symbol editing dialog.



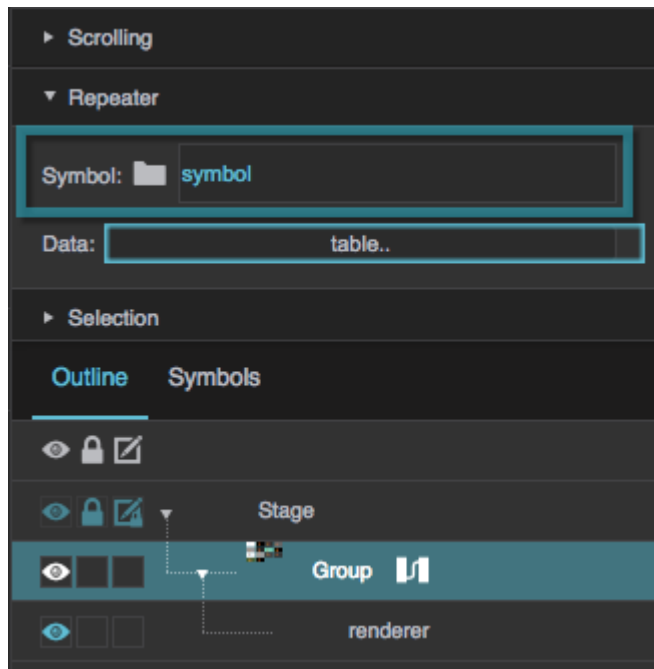
Finish the Repeater

1. In the Outline, select the group, and ensure that the group's **Layout** property is set to Horizontal or Vertical. You might also want to set the **Wrap** property to Wrap.

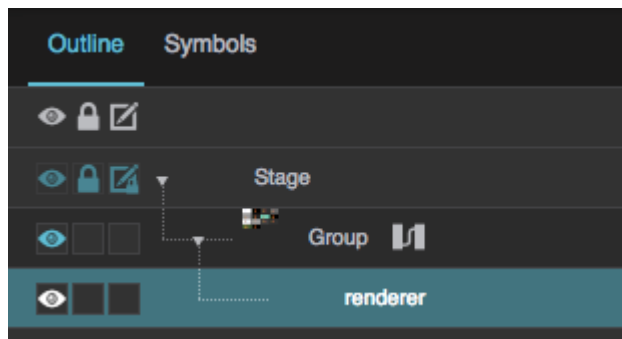
If the group's layout is Absolute, all the symbol instances will appear at the same location in the group.



2. In the Property Inspector, under **Repeater**, for **Symbol**, type symbol.

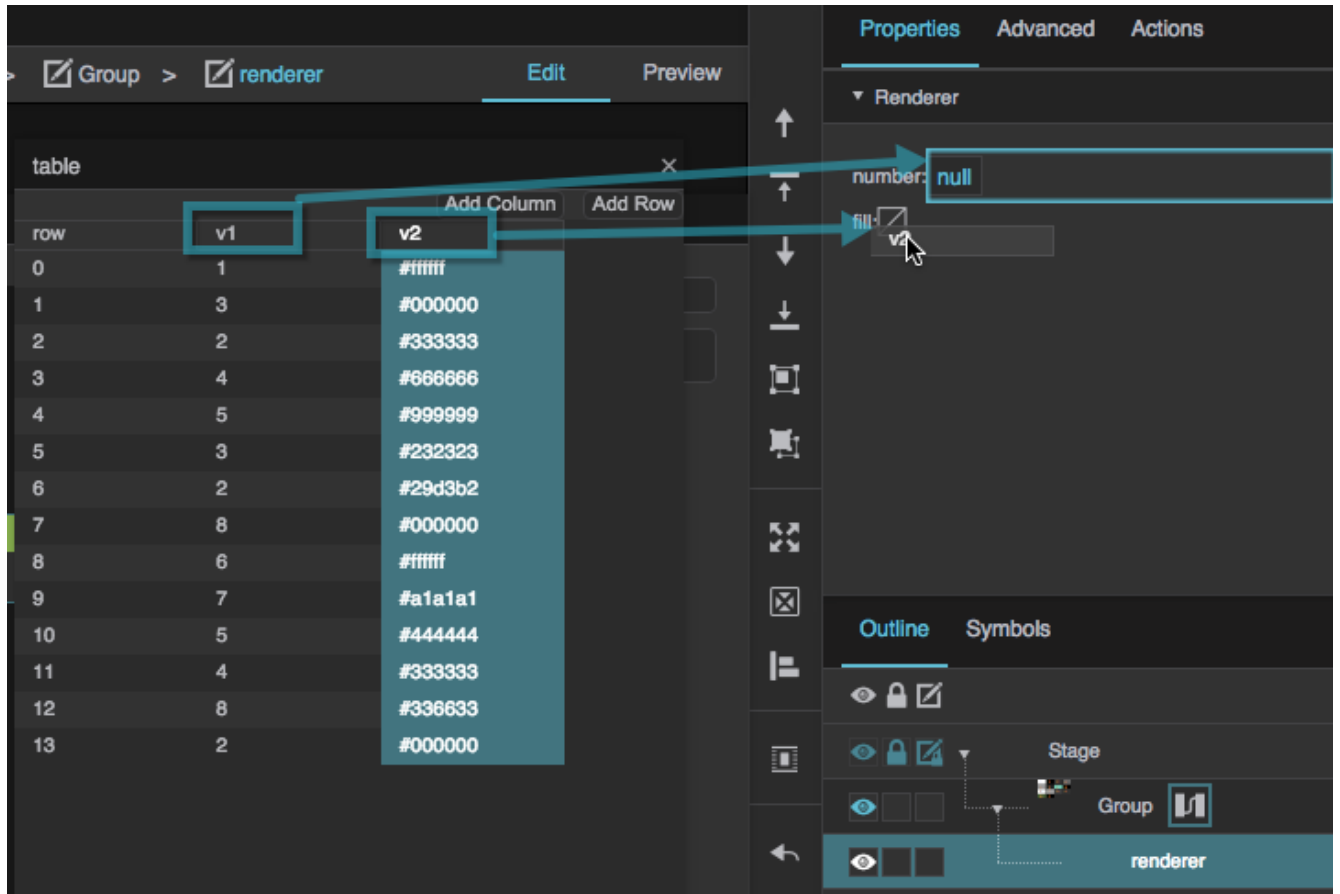


3. In the Outline, select the "renderer" node that is a child of the group.



4. With the "renderer" node selected, click **Dataflow** to open the dataflow for the group. Then, select the Table dataflow block, and click the button next to the **data** property to open the table.

1. Drag the v1 column header to the **number** property, and the v2 column header to the **fill** property.



You have created a repeater that shows one instance of the symbol for each row of data in the table.

1	3	2	4
5	3	2	8
6	7	5	4
8	2		

Configure Repeater Selection


You can control other components in your project based on the user's selection of items inside the repeater. To do this, you can use the group's [Selection properties](#). In this example, you will add a Single Select behavior to a repeater, and create a text component that displays the currently selected value. In these steps, you will use a [Table Row Cells](#) block.

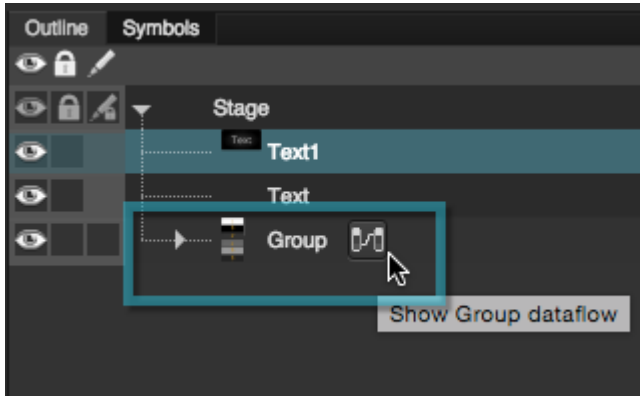
Follow these steps after you create a repeater in a group component.

In the same .dg5 file as your repeater:

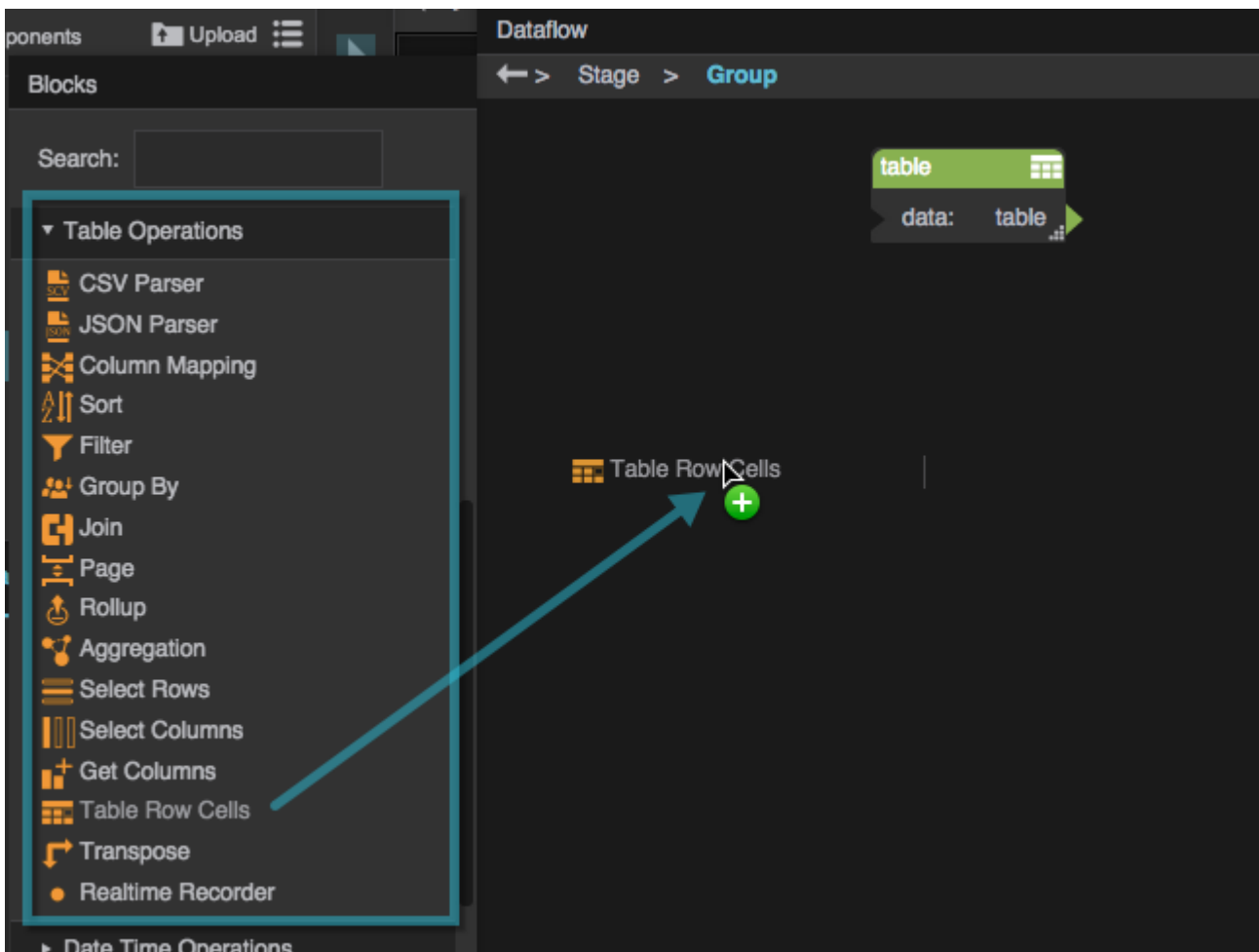
1. Right-click the Stage in the [Document window](#) or [Outline](#), and select **Insert > Components >**

Text.

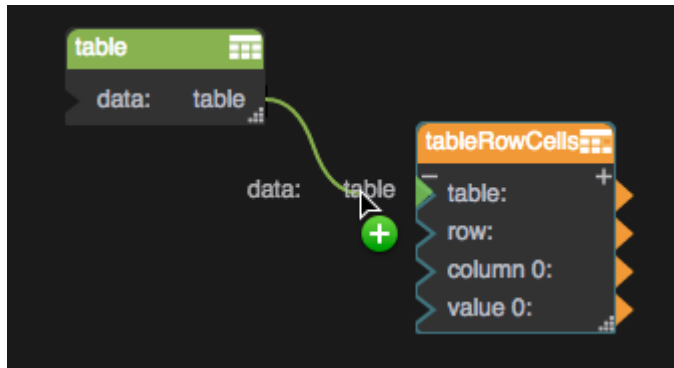
2. In the Outline, next to the group component that contains the repeater, click the  **Dataflow** icon.



3. In the Blocks Palette, expand **Table Operations**, and drag a **Table Row Cells** block to the dataflow.



4. Bind the table that is driving your repeater to the **input** property of the Table Row Cells block.



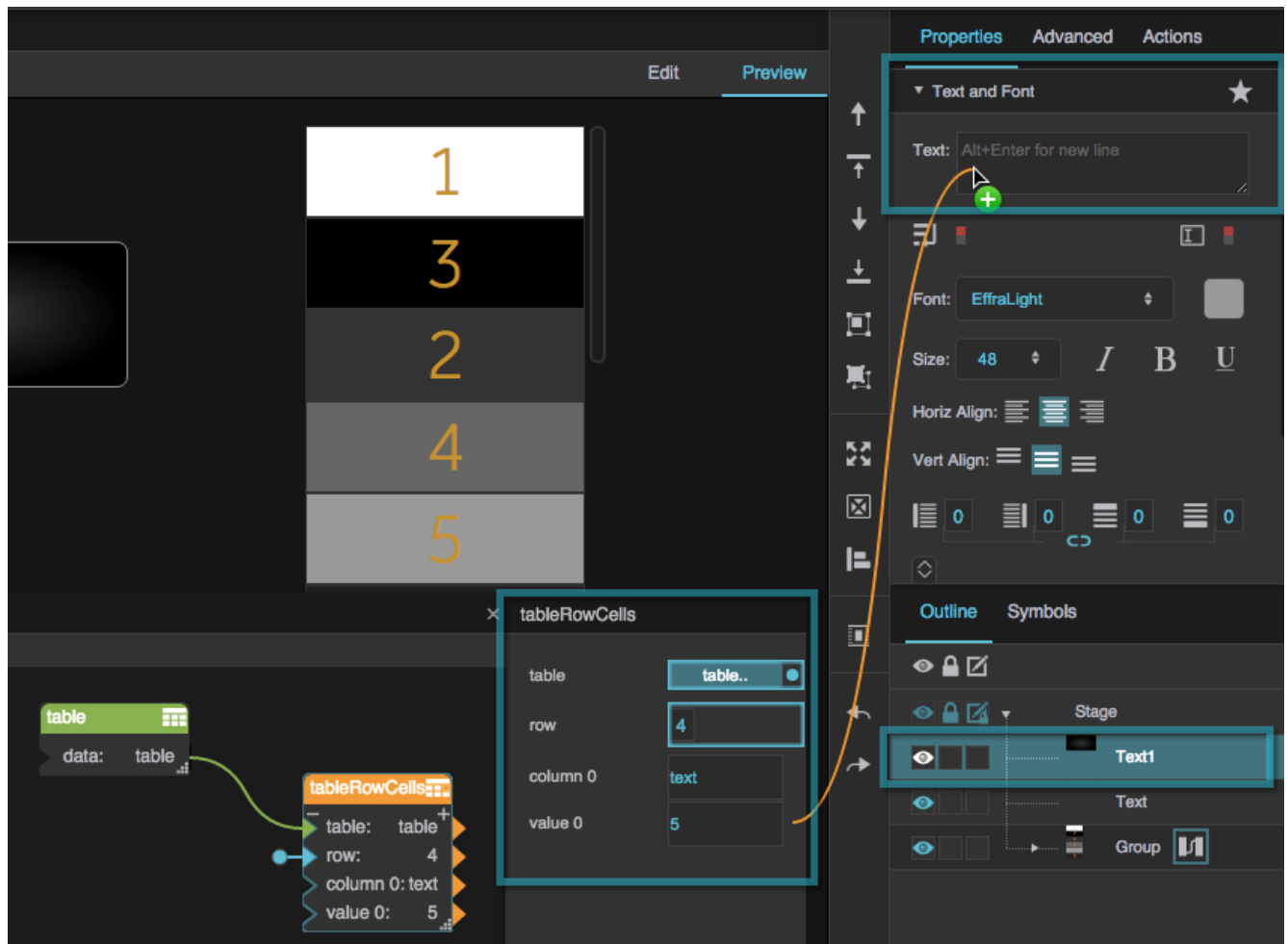
5. With the group selected in the Outline, expand the **Selection** properties, and set the Selection Behavior property to **Single Select**.
6. Bind the **Selected Index** property to the Table Row Cells block's **row** property.

7. For the Table Row Cell block's **column 0** property, enter the name of one of the columns from the table that is driving the repeater.

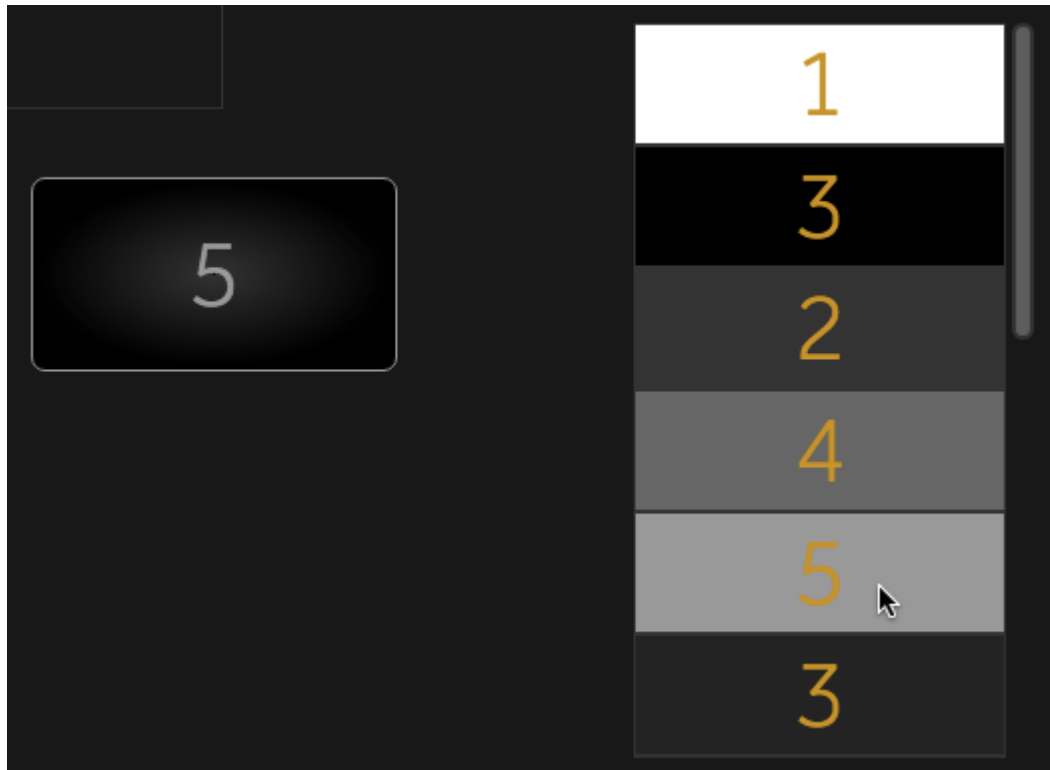
The screenshot shows the Dataflow IDE interface. On the left, a 'table' component is displayed with a table view. The table has two columns: 'row' and 'color'. The 'row' column contains values from 0 to 13, and the 'color' column contains various hex color codes. A 'tableRowCells' component is connected to the 'table' component. The 'tableRowCells' component has four properties: 'table' (set to 'table'), 'row' (set to '4'), 'column 0' (set to 'text'), and 'value 0' (set to '5'). The 'value 0' property is highlighted in blue, indicating it is selected. The 'table' component is also highlighted in blue, indicating it is selected.

If a repeater item is selected, the Table Row Cell block's **value 0** property automatically updates when you specify a **column 0**.

8. Without closing the dataflow, select the text component in the Outline.
9. Bind the Table Row Cell block's **value 0** property to the text component's **Text** property.




10. Change to [Preview mode](#), and test your repeater by clicking. The text component's contents should update depending on the items you click inside the group.

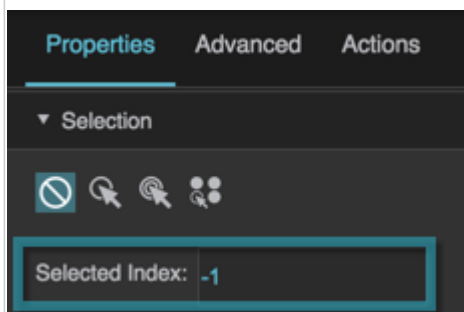


Repeaters FAQ

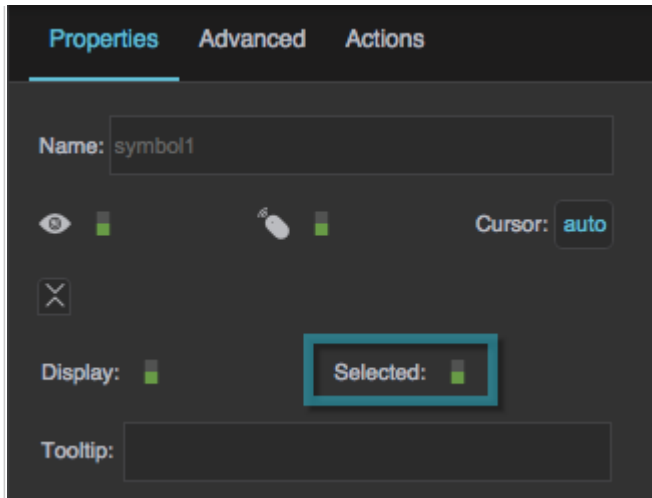
How do I create a repeater with multi-select behavior?

To create a repeater with multi-select behavior, follow the steps above to create a repeater. Then, select the group that contains the repeater, and in the group's [Selection properties](#), for **Selection Behavior**, choose  **Multi-Select**. Be sure to change to [Preview mode](#) when you test the behavior.

When items in the repeater are selected, their indexes appear in the **Selected Index** property as comma-separated values.



Also, the **Selected** property of selected symbol instances is set to TRUE.

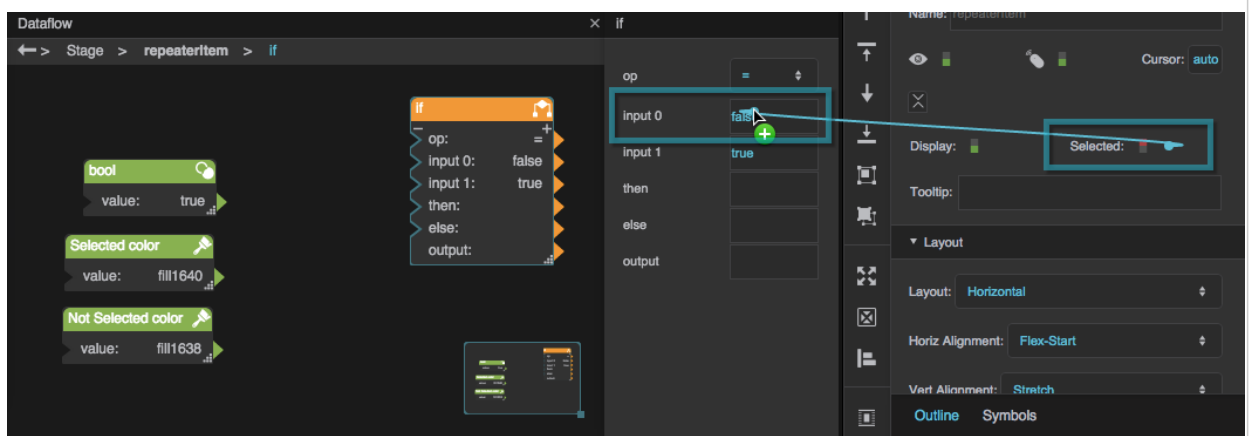


How do I change the appearance of items in my repeater depending on whether they are selected?

These steps show you how to use the **Selected** property of the repeater's symbol in order to control repeater item background color. Follow these steps after you create a repeater in a group component. In this example, you will use **If**, **Boolean**, and **Color** dataflow blocks.

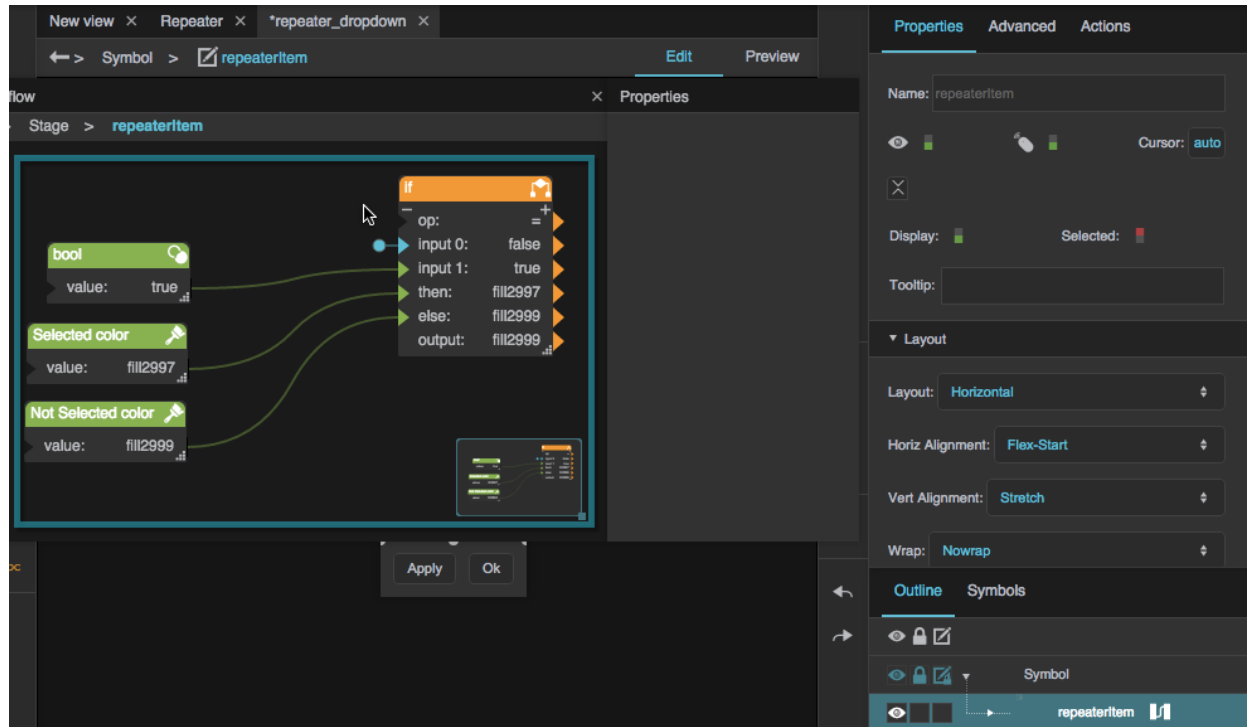
In the same `.dg5` file as your repeater:

1. In the **Symbols Panel**, right-click the symbol that represents an item in your repeater, and select **Edit**.
2. In the **Outline**, right-click the symbol, and select **Dataflow**.
3. In the dataflow Blocks palette:
 1. Expand **Logic**, and drag an **If** block to the dataflow window.
 2. Expand **Variables**, and drag a **Boolean** block and two **Color** blocks to the dataflow window.
4. For the Color blocks that you added, set one of the values to be a "Selected" fill and the other to be a "Not Selected" fill.
5. For the Boolean block that you added, set the value to TRUE.
6. In the If block that you added:
 1. For **input 0**, drag the **Selected** property of the symbol.

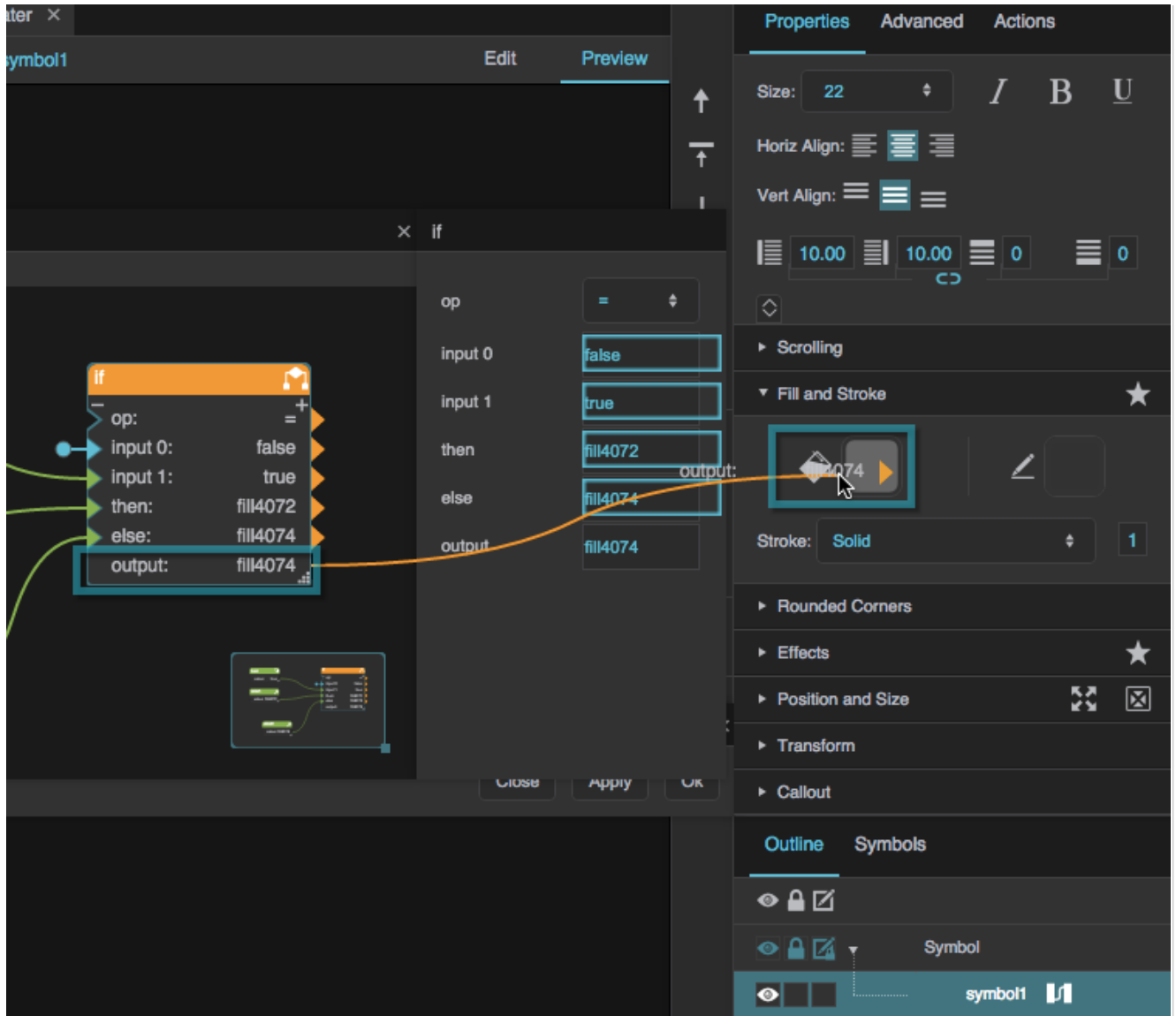


2. For **input 1**, bind the TRUE value from the Boolean block.

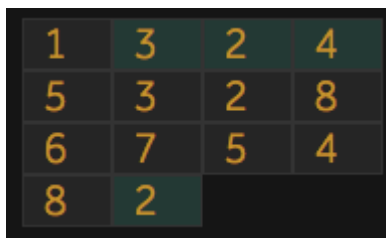
3. For **then**, bind the "Selected" fill.
4. For **else**, bind the "Not Selected" fill.



7. Bind the **output** property of the If block to the **Fill** property of the symbol.



- Close the dataflow window, and click **OK** to close the symbol editing dialog.
- In **Preview mode**, test your repeater. Items in the repeater change color based on whether they are selected.



Users will scroll inside my repeater using touch screens. What should I do?

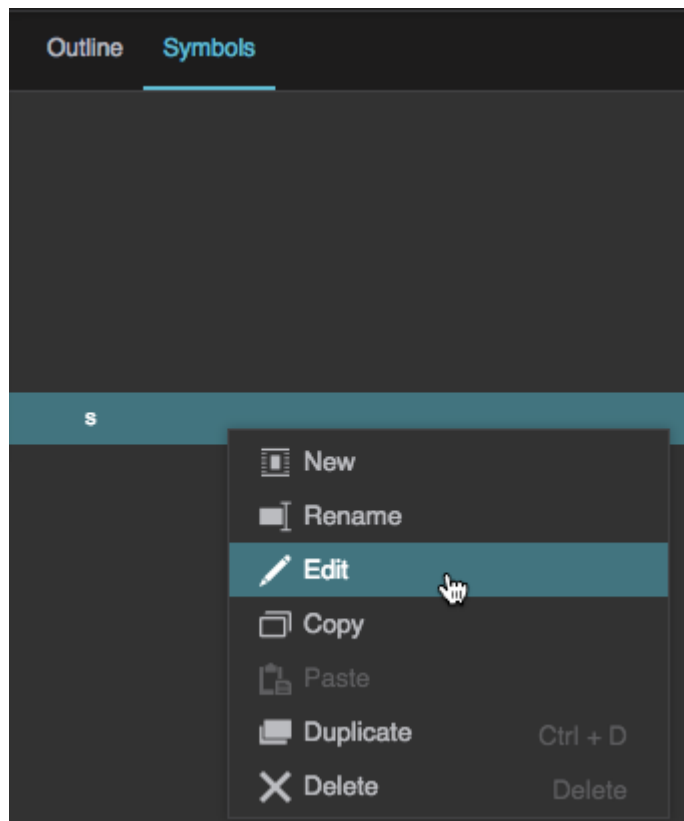
If users will scroll inside your repeater using touch screens, the `onClick` event that typically drives selection should not be used because the act of scrolling can trigger selection. Instead, you can use the `onMouseUp` event to drive selection. To do this, you can create an [action](#) that affects the repeater's

source symbol. The action drives the symbol's [Selected](#) property. In this example, you will create a text component that displays a value from a column in the repeater's table.

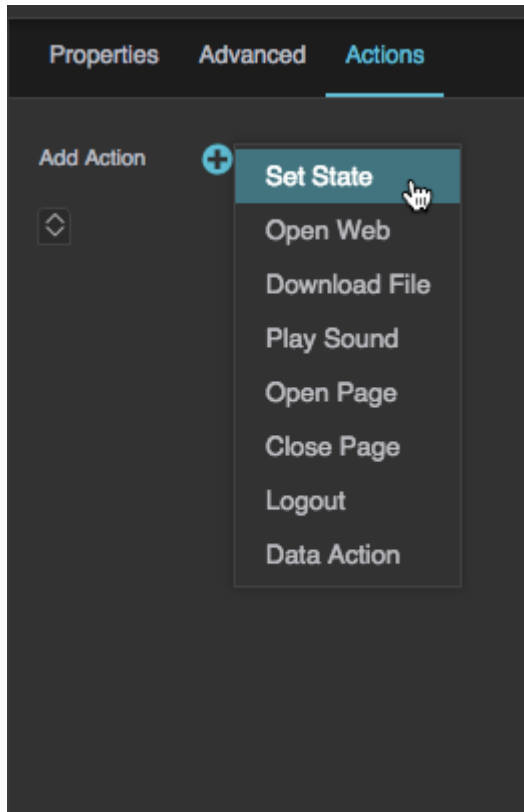
Before following these steps, create a repeater as described above, without selection behavior.

In the same `.dg5` file as your repeater:

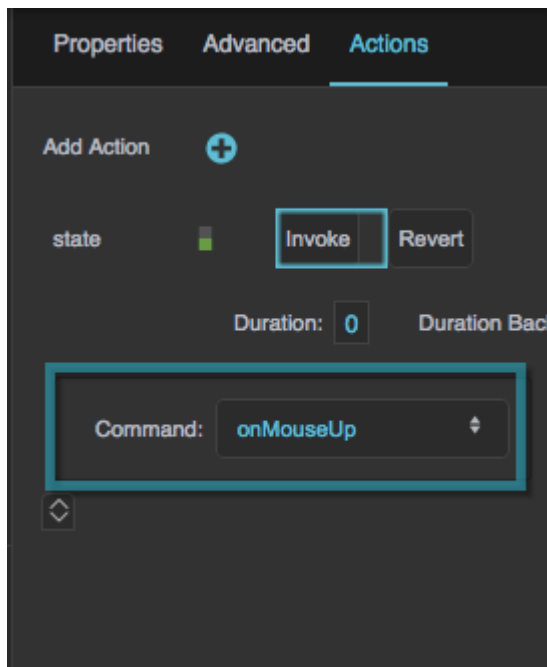
1. In the [Symbols panel](#), right-click the symbol that represents an item in your repeater, and select **Edit**.



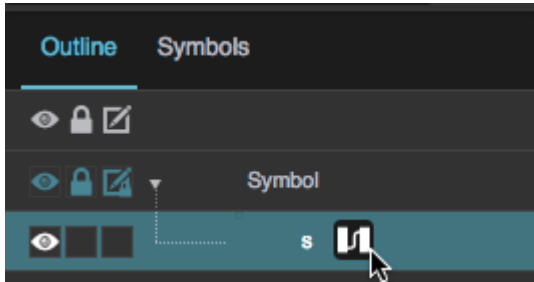
2. Create an action:
 1. Click the **Actions** tab to open the [Actions panel](#).
 2. Click **+ Add Action**, and select **Set State**.



3. For the action's **Command Type**, select **onMouseUp**.

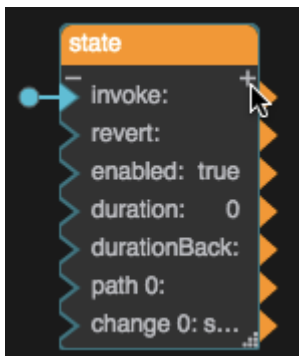


4. Open the Outline, and next to the symbol, click the **Dataflow** icon.

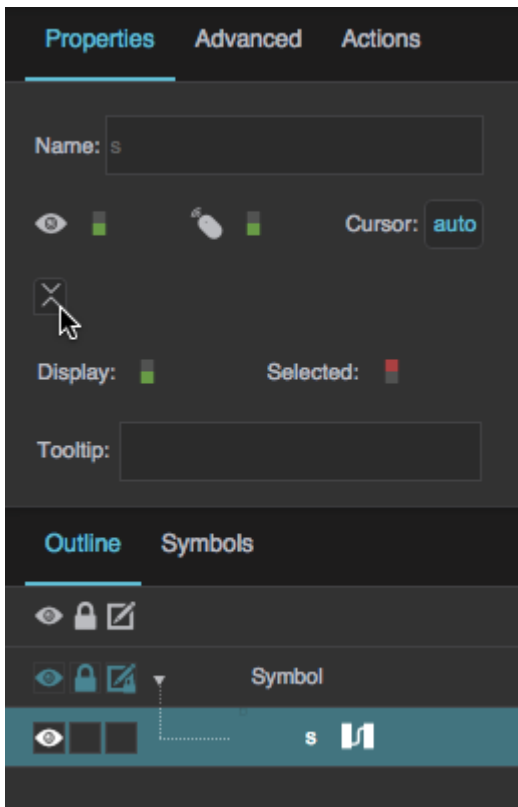


5. In the State block, click the plus sign.

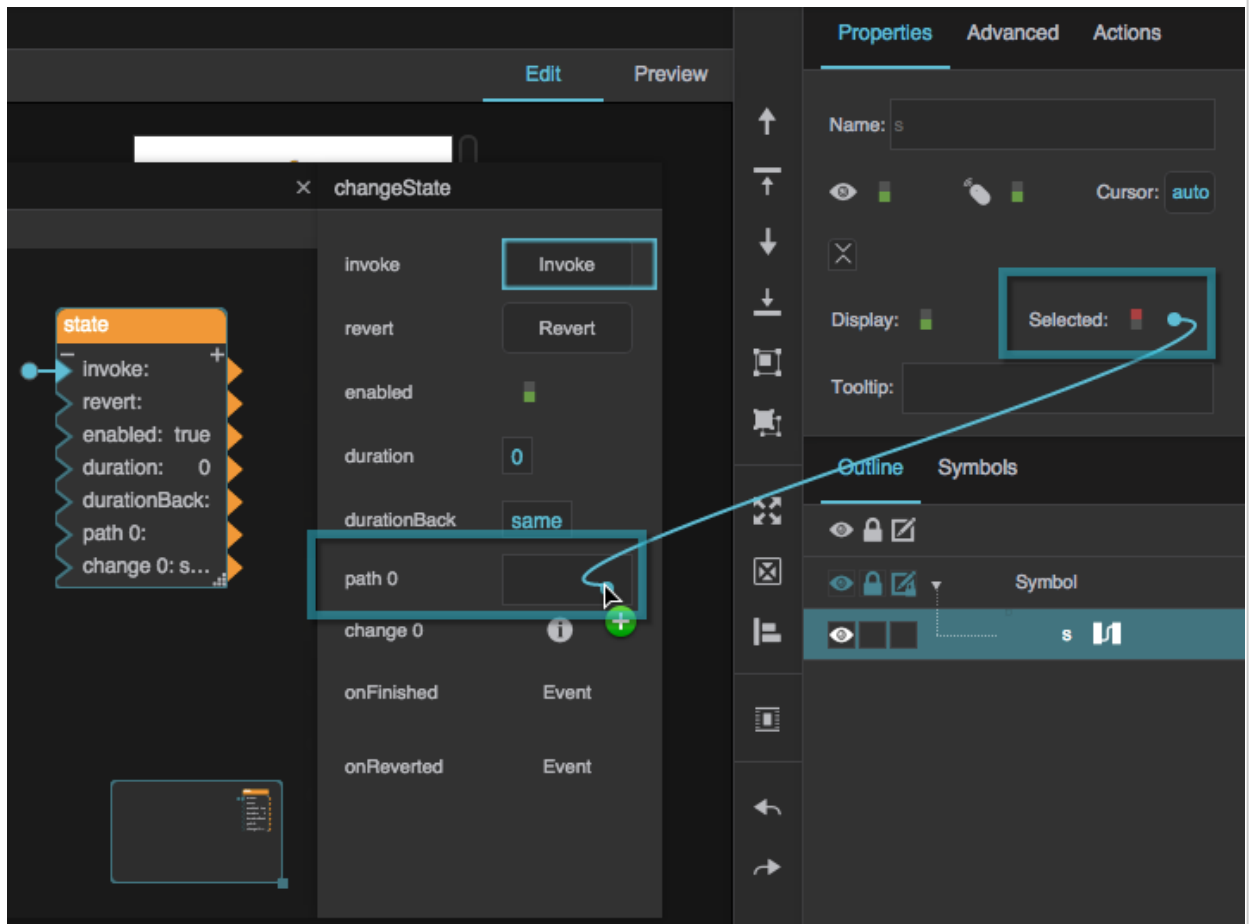
This adds **path 0** and **change 0** properties.



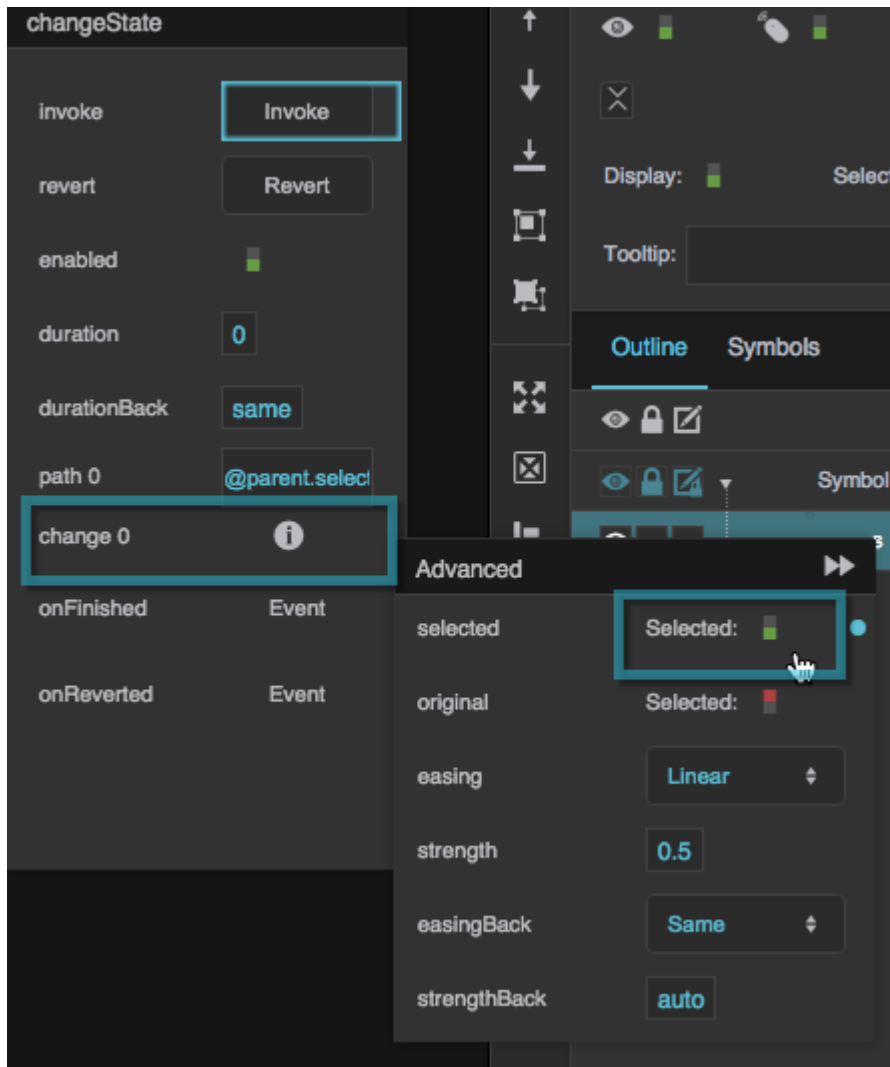
6. Without closing the dataflow, open the Property Inspector, and expand the top section of properties.



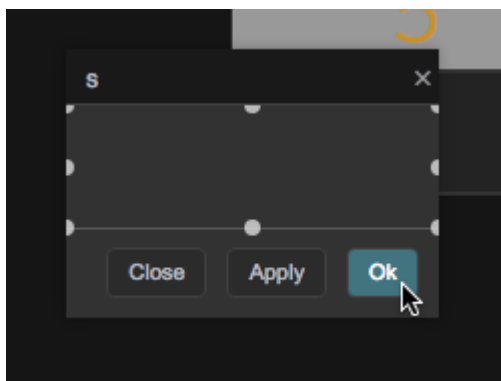
7. Bind the symbol's **Selected** property to the State block's **path 0** property.



8. For the State block's **change 0** property, click the **i** icon, and set **Selected** to TRUE.

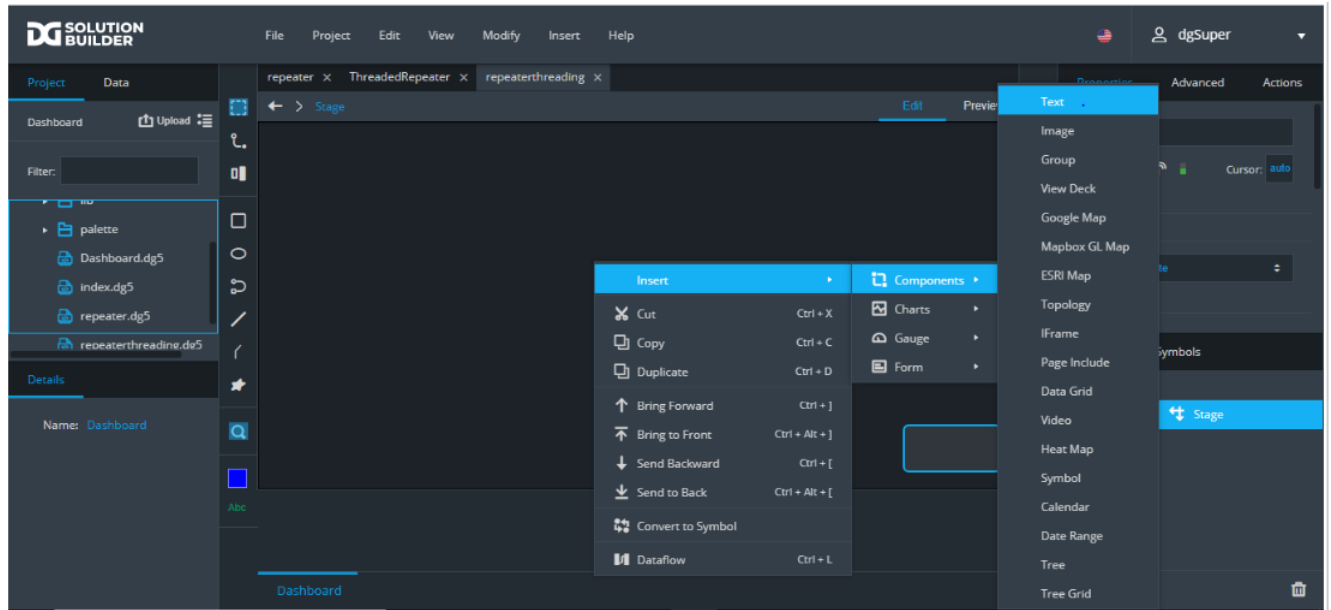



3. Click **OK** to exit the symbol editing dialog.

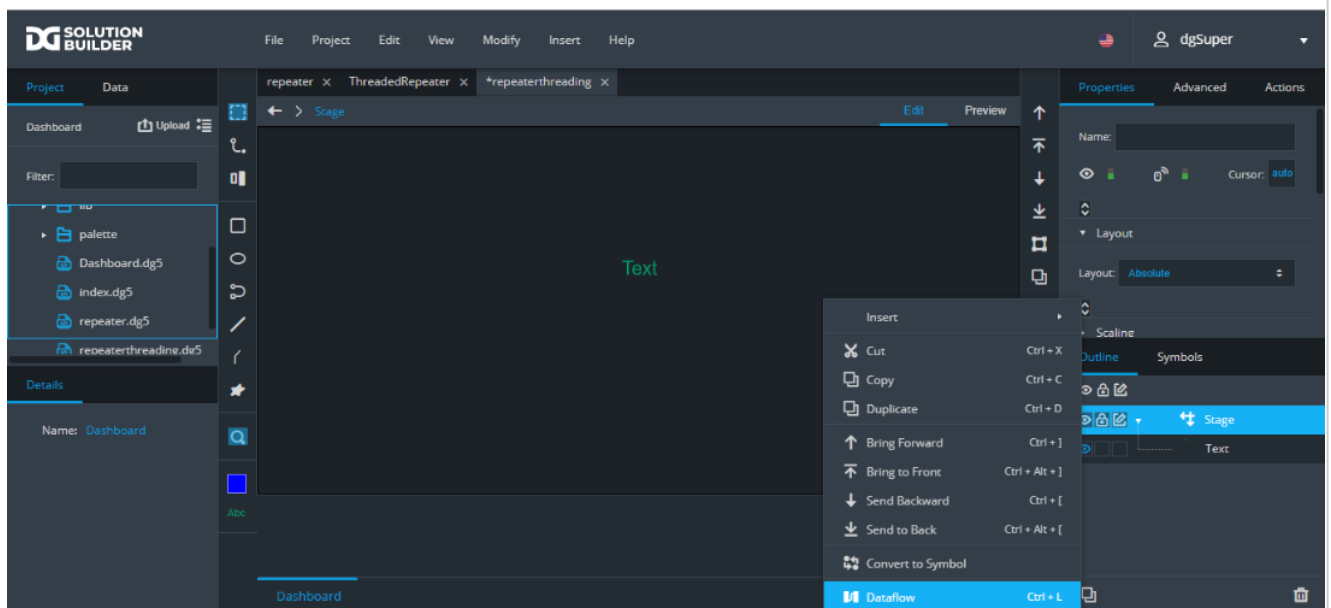


You have created custom selection behavior that uses the onKeyUp event instead of the onClick event.

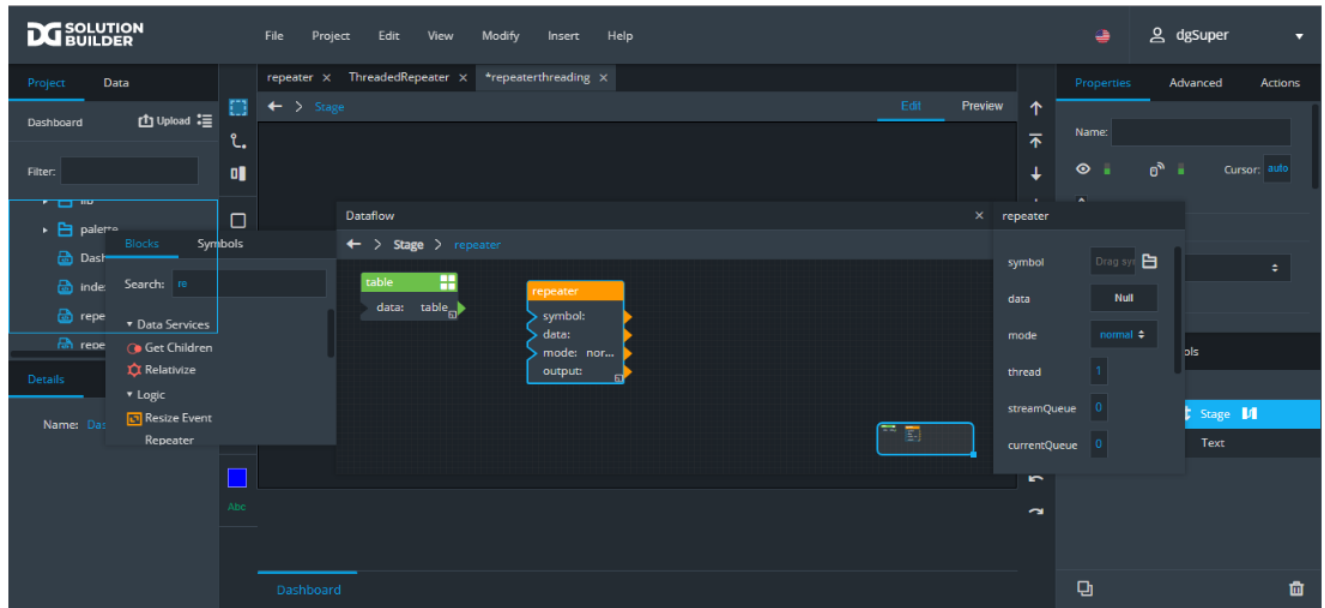
1. In the **Outline panel**, right-click the **Stage** and select **Insert > Components > Text**.




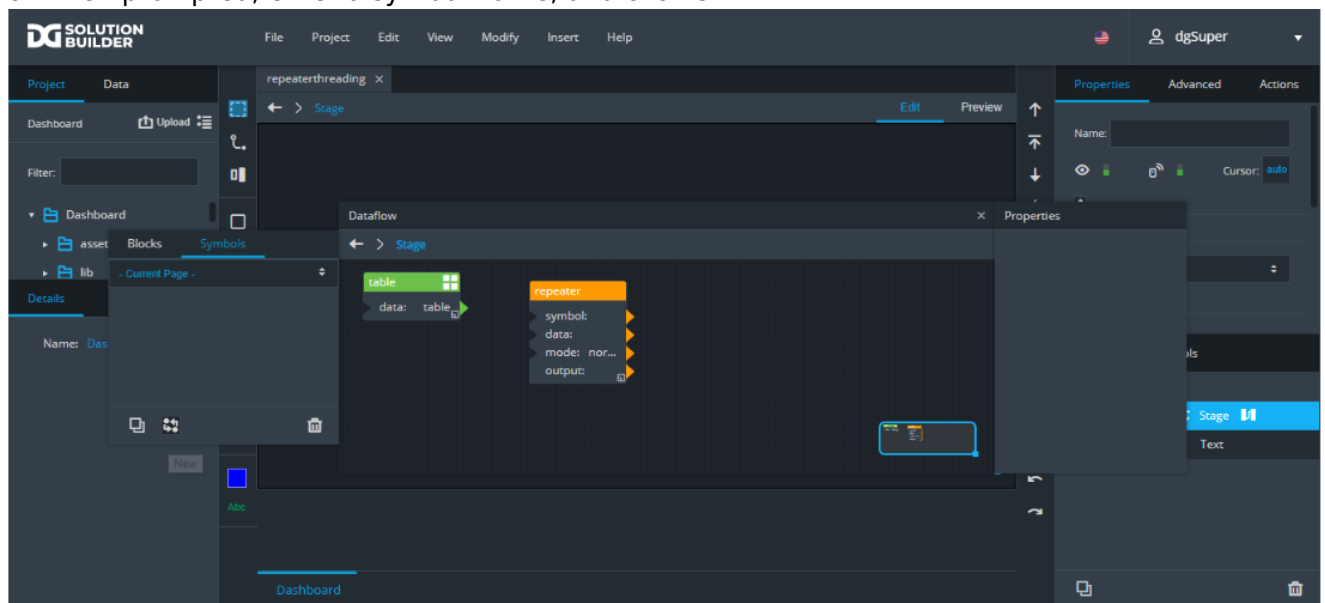
2. Right-click the **Stage** and click  **Dataflow** icon.



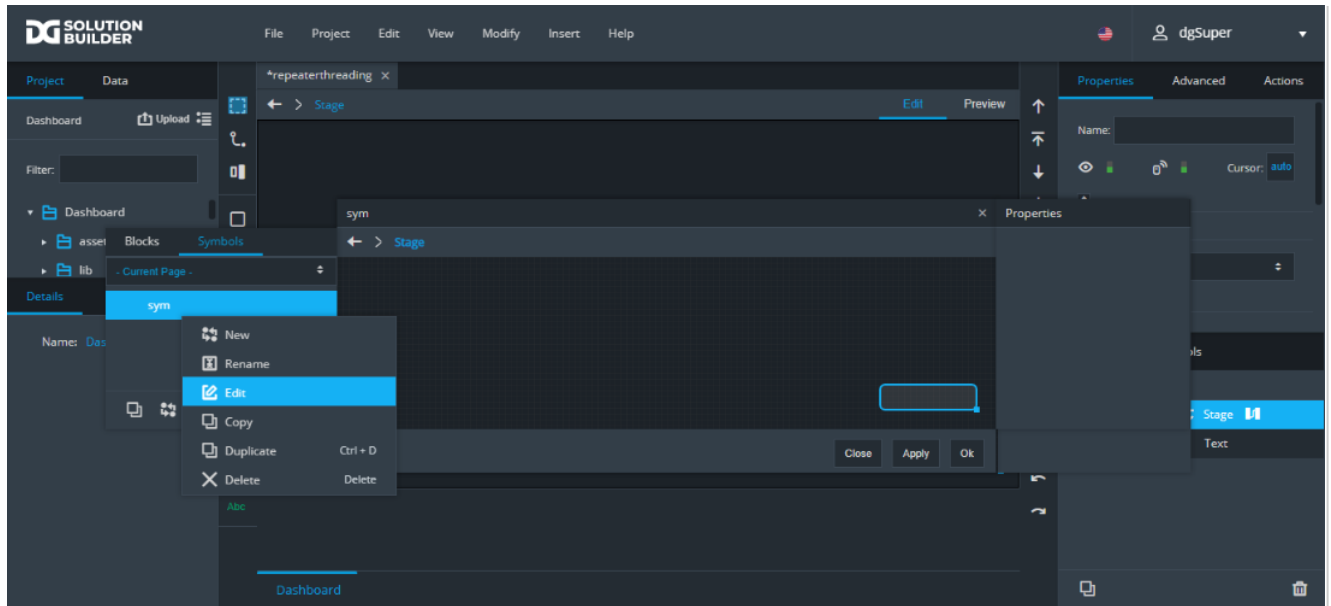
3. Drag a **Table** dataflow block and a **Repeater** block to the dataflow window.



4. To create a new dataflow symbol and to enter edit mode for that symbol:
 - a. Switch to the **Symbols** panel.
 - b. Click the  **Convert to symbol** icon.
 - c. When prompted, enter a symbol name, and click OK.

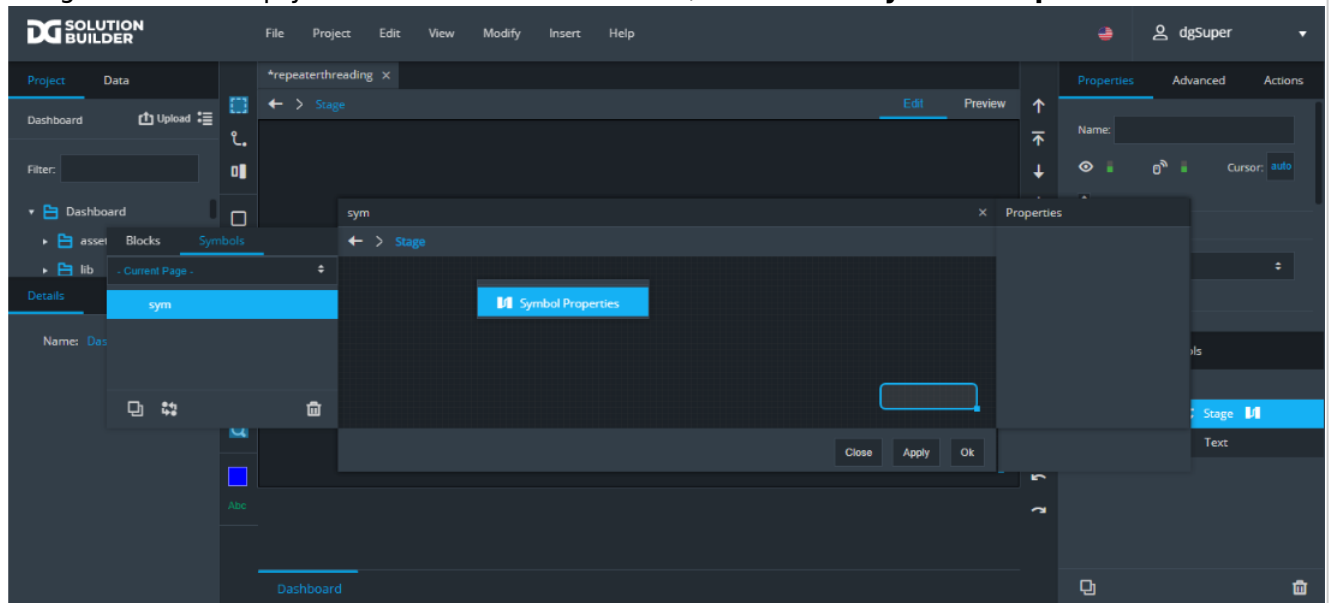


- d. Right-click a symbol instance in the dataflow window, and choose **Edit**.

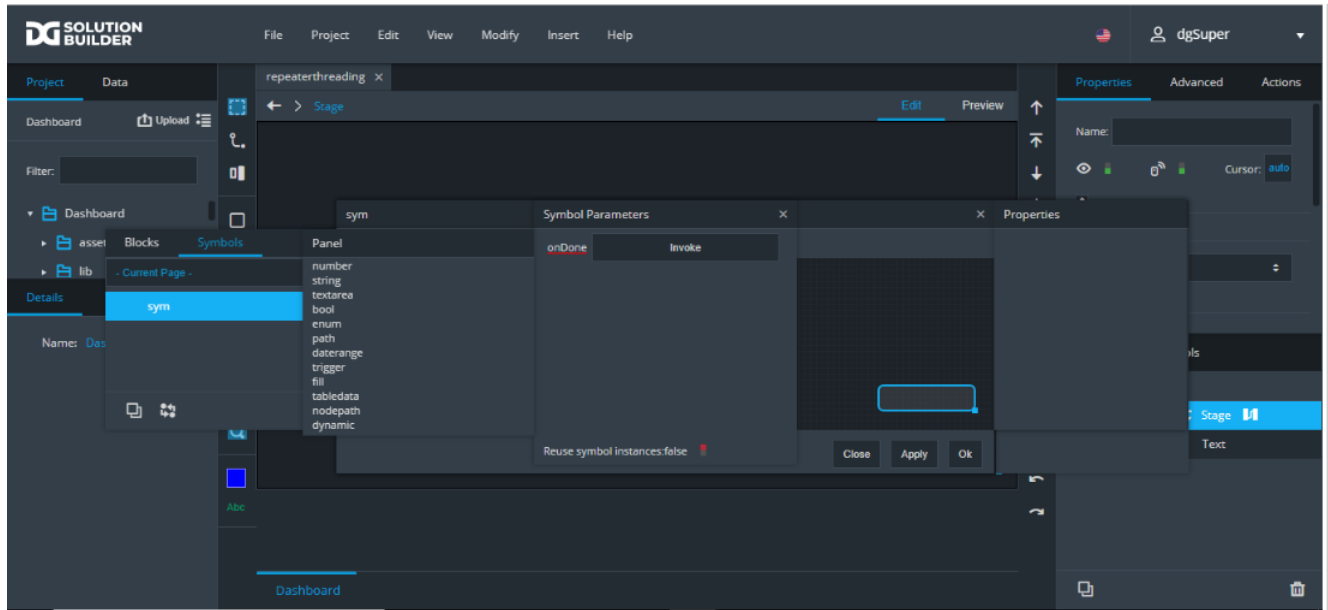


e. Edit the dataflow model according to the requirements.

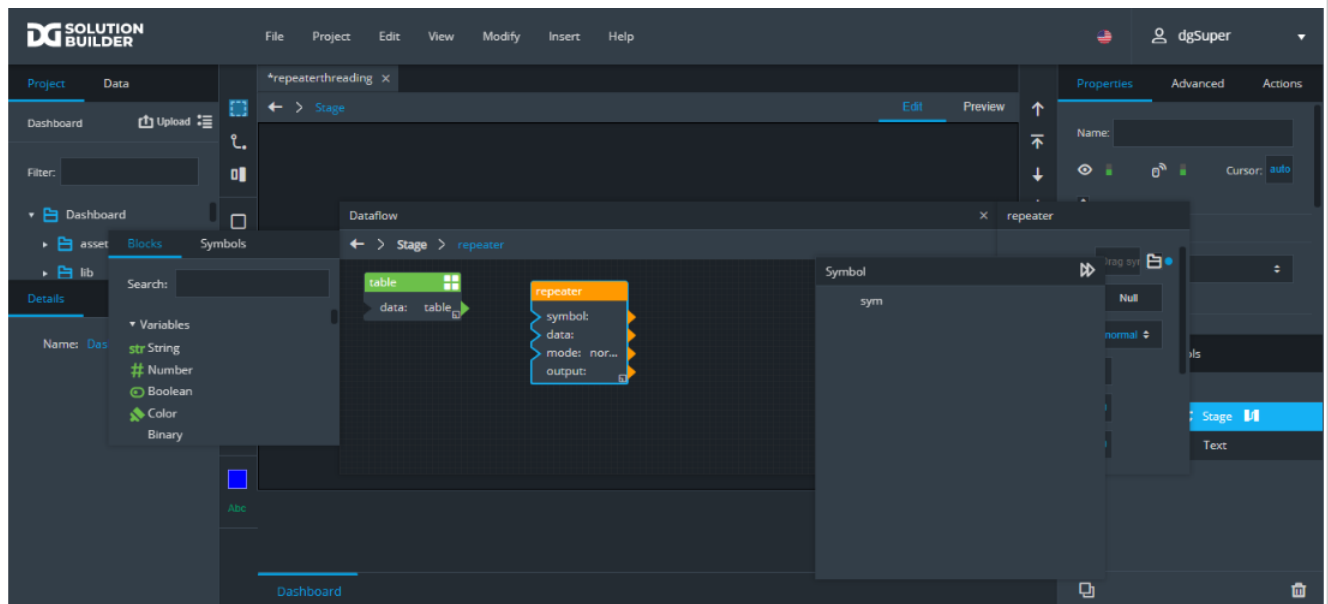
f. Right-click an empty area in the dataflow window, and choose **Symbol Properties**.



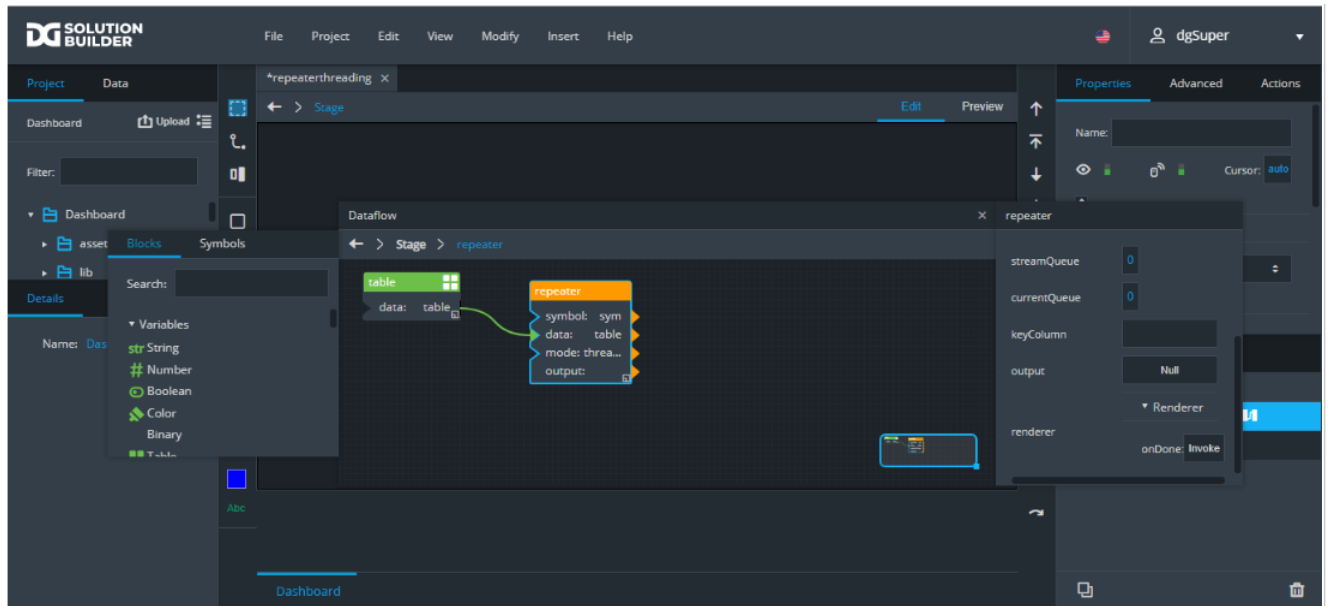
g. Drag **trigger** from **Panel** to **Symbol Parameters**. Rename trigger to **onDone**.



5. In the **Repeater** block, browse for the symbol in block's parameters.



6. Bind table to the **data** field of the **Repeater** block.



Now you can trigger other actions using **onDone** property.

ThreadedRepeater.dg5

```

{"@type": "group", "Text": {"@type": "text", "width": 504, "cornerRadius": 0, "multiline": true, "fontSize": "12", "height": 185, "border": {"@type": "fill", "gradient": {"@type": "gradient", "@array": [{"ratio": 0, "color": "#000000", "alpha": 1}, {"ratio": 100, "color": "#ffffff", "alpha": 1}]}, "color": "16777215", "text": ["@parent.storage.value"], "color": "12303296", "left": -5.684341886080802e-14, "top": 5.595524044110789e-14, "storage": {"@type": "localStorage", "!ps": [{"key", "value"}, {"!x": 444, "!y": 155, "key": "debug_out", "value": "", "!w": 208}, "width": "100%", "height": "100%", "clipContent": true, "repeater": {"@type": "repeater", "render": {"map": {"string": ["@parent.@data.v1"]}}, "!ps": [{"symbol", "data", "mode", "output"}, {"!x": 293, "!y": 34, "symbol": "s", "data": ["@parent.table.data"], "mode": "threaded"}, "table": {"@type": "table", "data": {"@type": "table data", "cols": [{"name": "row", "type": "number", "meta": {"name": "row", "type": "number"}}, {"name": "v1", "meta": {"name": "v1"}}, "rows": [[0, "value 1"], [1, "vvv 2"], [2, "last row"]]}, "!ps": [{"data"}, {"!x": 46, "!y": 37}, {"!df": true, "@array": [{"Text"}], "@symbols": {"@df": {"s": {"@type": "dfcontent", "script": {"@type": "script", "!ps": [{"invoke", "script", "output", "print", "a", "onDone"}, {"!x": 166, "!y": 63, "invoke": ["@parent.@params.onInit"], "script": "@parent.storage.value = @parent.storage.value + \\\"\\n triggered: \\\" + @.a;\\n@.onDone = Math.random();", "!var": [{"n": "a", "t": "string"}, {"n": "onDone", "t": "trigger"}], "a": ["@parent.@params.string"], "!df": true, "storage": {"@type": "localStorage", "!ps": [{"key", "value"}, {"!x": 378, "!y": 82, "key": "debug_out", "!w": 208}, "@params": {"!var": [{"n": "onInit", "t": "trigger"}, {"n": "onDone", "t": "trigger"}, {"n": "string", "t": "string"}], "onDone": ["@parent.script.onDone"]}}}}}, {"@ver": 8503}

```

[Previous: List Input](#)

[Next: Callouts](#)

From:

<https://wiki.dglogik.com/> - **DGLogik**

Permanent link:

https://wiki.dglogik.com/dglux5_wiki:widgets_and_property_inspector:repeaters:home

Last update: **2021/09/20 14:51**

