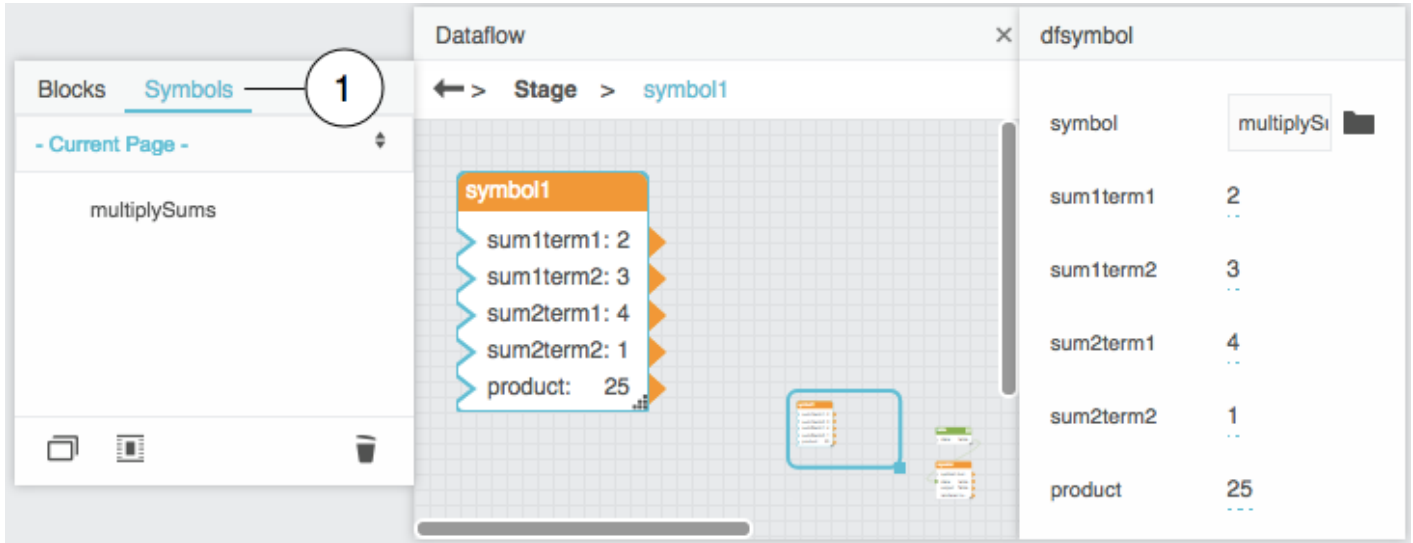


Dataflow Symbols and Dataflow Repeaters

This page describes dataflow symbols and dataflow repeaters in DGLux5 and includes basic operations and tutorials.

A *dataflow symbol* is a [dataflow model](#) without a parent object. You access existing dataflow symbols in the *symbols panel* of the dataflow. The symbols panel is located in a tab next to the block palette. The following image shows the location of the symbols panel.



1 Symbols panel

After you create a dataflow symbol, you can add *instances* of that symbol to dataflow models. A symbol instance is a single dataflow block that represents the symbol. Instances of a symbol are identical to one another except where affected by symbol *parameters* that you create. If a symbol does not have parameters, all of its instances are identical.

You can edit a symbol, or you can edit a symbol instance. Changes to a symbol affect all instances of the symbol. Changes to a symbol instance affect only that instance. Only the following aspects of a symbol instance can be edited.

- Position and size
- [Page code name](#) and block label
- Which properties are pinned and unpinned
- Parameter values

A *dataflow repeater* is a [Repeater](#) block. A Repeater block takes a table and a dataflow symbol as input. Then, the Repeater block returns an output table that describes parameter values of the input symbol for each row of the input table.

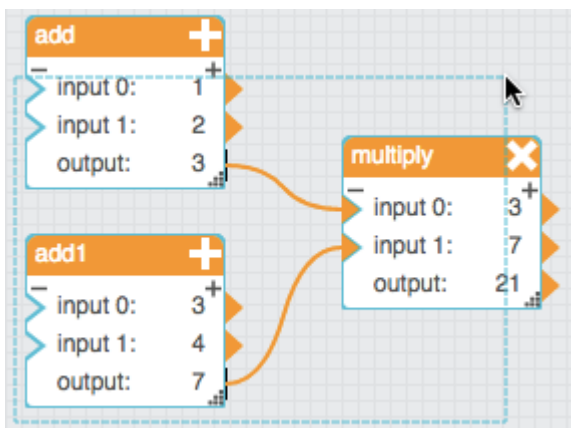
How to Create a Dataflow Symbol

You can create a new dataflow symbol in two ways.

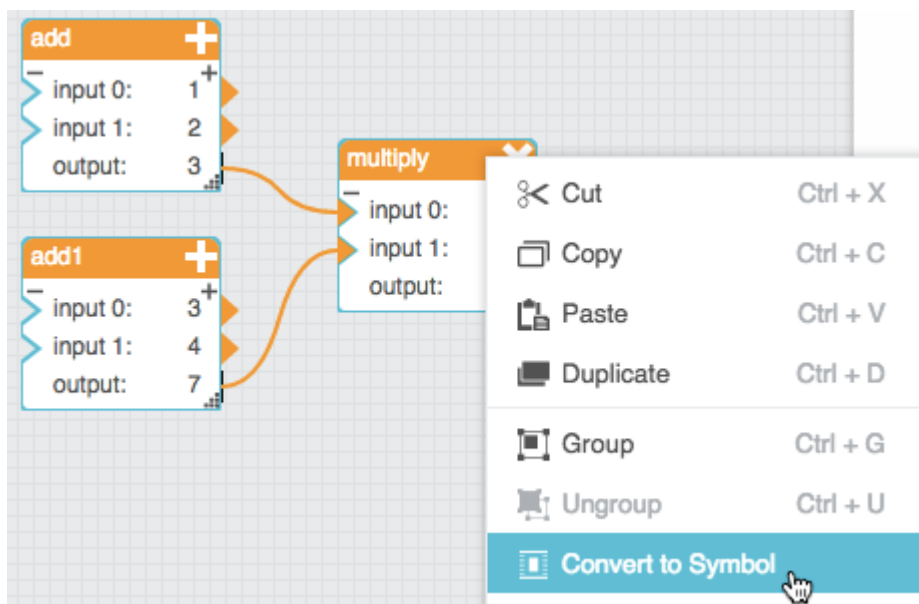
How to Create a New Dataflow Symbol via Block Conversion

To create a new dataflow symbol:

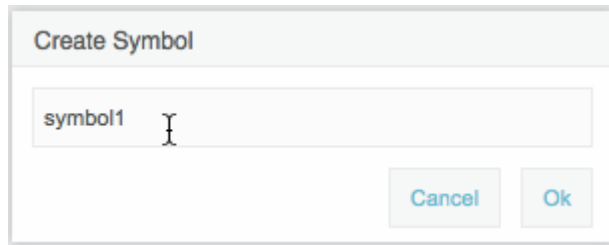
1. Add blocks to any dataflow model.
2. Select the blocks to convert.



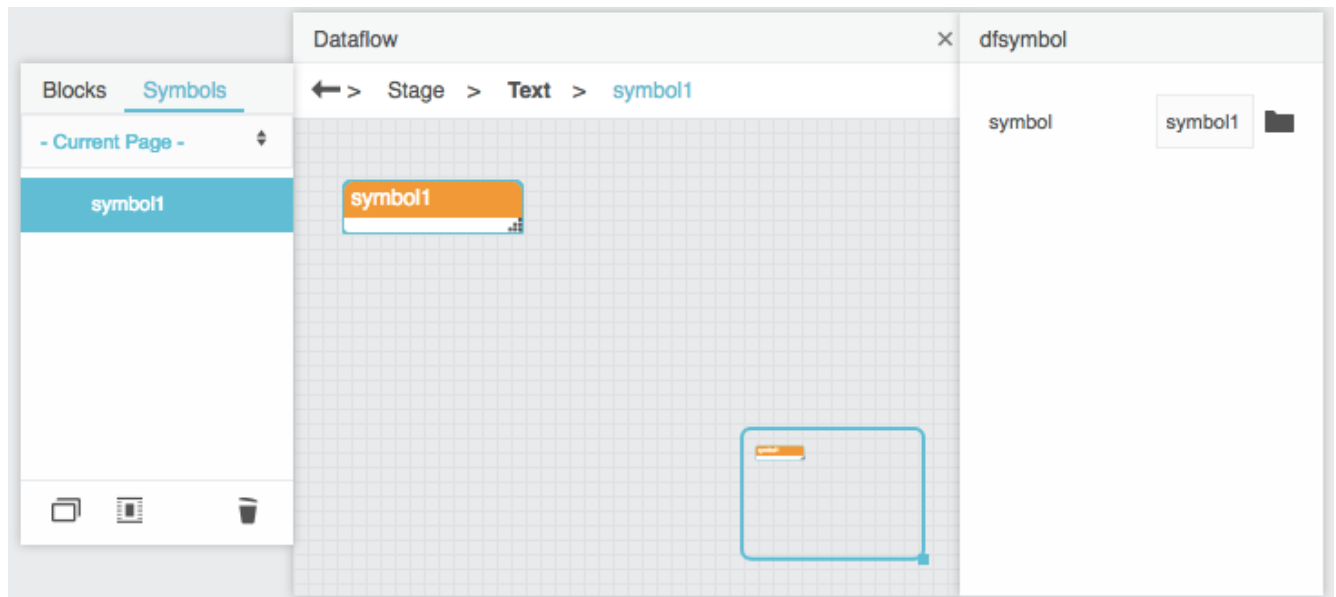
3. Right-click one of the selected blocks, and choose **Convert to Symbol**.



4. When prompted, enter a symbol name, and click **OK**.




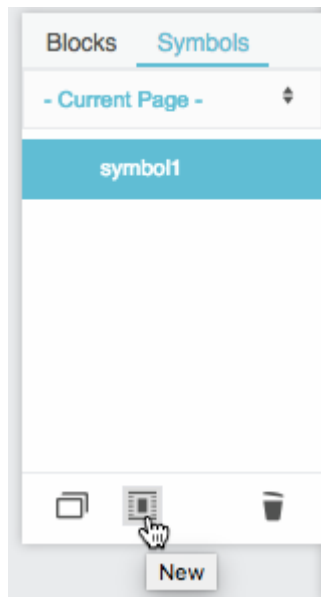
In the current dataflow model, the selected blocks are replaced by a symbol instance. In the symbols panel, the new symbol appears in the symbols list.



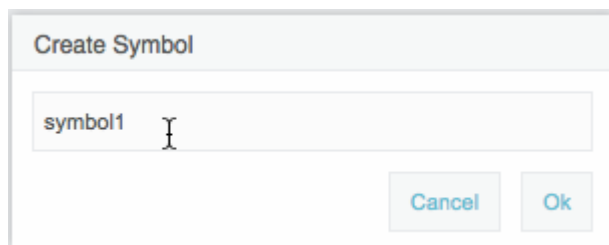
How to Create a New Dataflow Symbol via the Symbols Panel

To create a new dataflow symbol and also enter symbol editing mode for that symbol:

1. Make sure the symbols panel, not the block palette, is selected.
2. Click the  **New** icon.



3. When prompted, enter a symbol name, and click **OK**.



How to Edit a Dataflow Symbol

You edit a dataflow symbol in *symbol editing mode* for that symbol. When you are in symbol editing mode, the dataflow view contains the dataflow model for the symbol, and three buttons appear that let you close, apply, or save the current dataflow model. These buttons appear only in symbol editing mode. The following image shows symbol editing mode.

1 Symbol editing mode buttons

If you are in symbol editing mode for multiple symbols simultaneously, the dataflow view for each symbol appears in a separate floating panel.

While in symbol editing mode, you can make two kinds of changes. The first kind of change affects the symbol dataflow model. The second kind of change exposes and manages symbol parameters.

Symbol parameters appear as block properties for symbol instances. These properties behave like other dataflow block properties, in that the property values can be different for each symbol instance. All symbol instance properties are unpinned by default.

Notes




- You always must click **OK** to save changes. Changes to a dataflow symbol are not saved automatically when you leave symbol editing mode.
- While you are in symbol editing mode for a dataflow symbol, you can click **Apply** to temporarily apply changes to all symbol instances. If you click **Apply** but do not click **OK**, your changes are lost when you exit symbol editing mode.

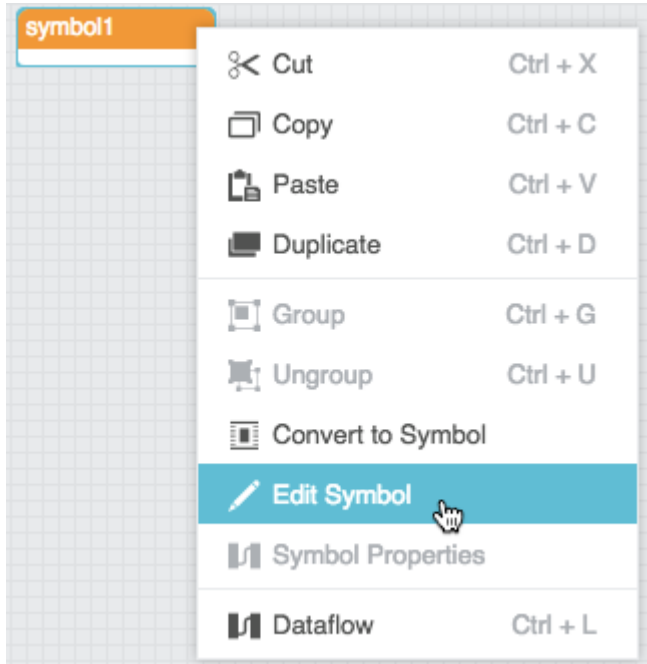
How to Enter Symbol Editing Mode

The following interactions enter symbol editing mode for a dataflow symbol:

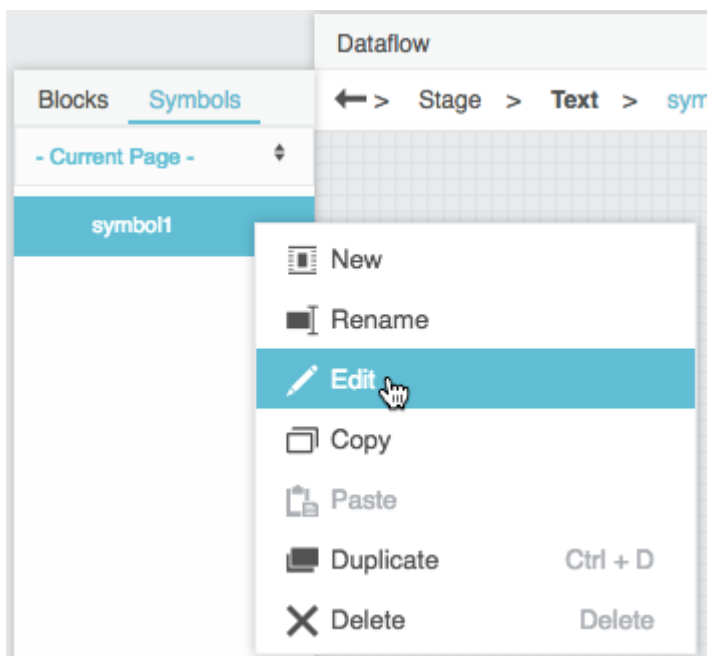
- Right-click a symbol instance in the dataflow window, and choose **Edit Symbol**.
- Right-click a symbol in the symbols panel, and choose **Edit**.

- In the symbols panel, click the  **New** icon, and then specify a name for the new symbol.

The following image shows how to enter symbol editing mode via the dataflow window.



The following image shows how to enter symbol editing mode via the symbols panel.



How to Edit a Symbol Dataflow Model

Changes made to a symbol dataflow model affect all instances of that symbol.

To edit a symbol dataflow model:

1. Make sure you have entered symbol editing mode.
2. Edit the dataflow model.

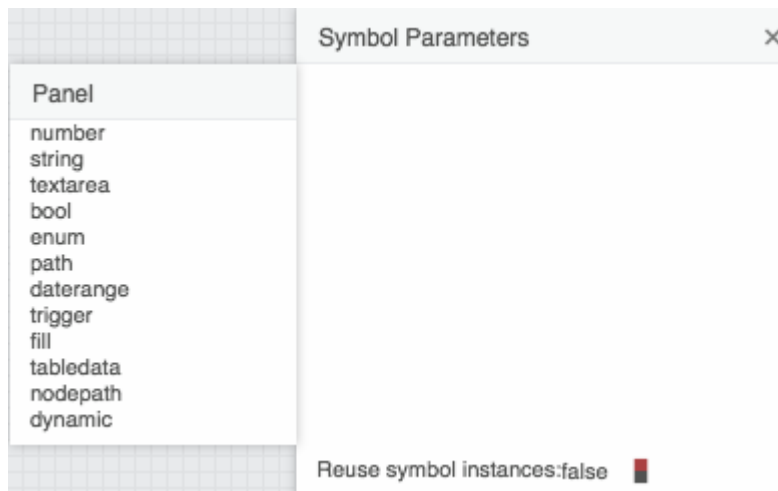
For example, you can do the following:

- Add blocks.
 - Rename, move, and delete blocks.
 - Edit block properties.
 - Create and delete bindings.
3. If you want to temporarily apply your changes to symbol instances, click **Apply**.
 4. Always click **OK** to save changes.

How to Edit Symbol Parameters

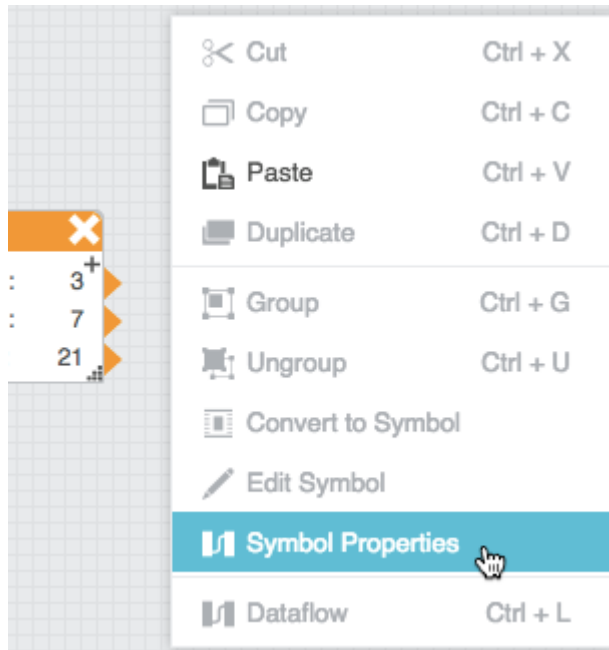
Symbol parameter values can be different for each symbol instance.

You create parameters using a Symbol Parameters pop-up. The leftmost portion of the Symbol Parameters pop-up contains data types that you can add to a symbol. The rightmost portion of the Symbol Parameters pop-up contains the parameters of the current symbol. The following image shows the two portions of the Symbol Parameters pop-up.



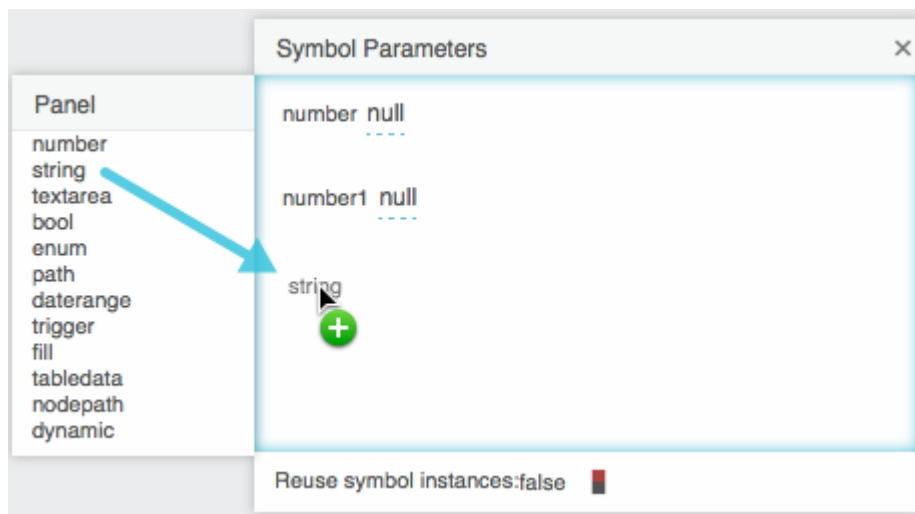
To define parameters:

1. Make sure you have entered symbol editing mode.
2. Right-click an empty area in the dataflow window, and choose **Symbol Properties**.

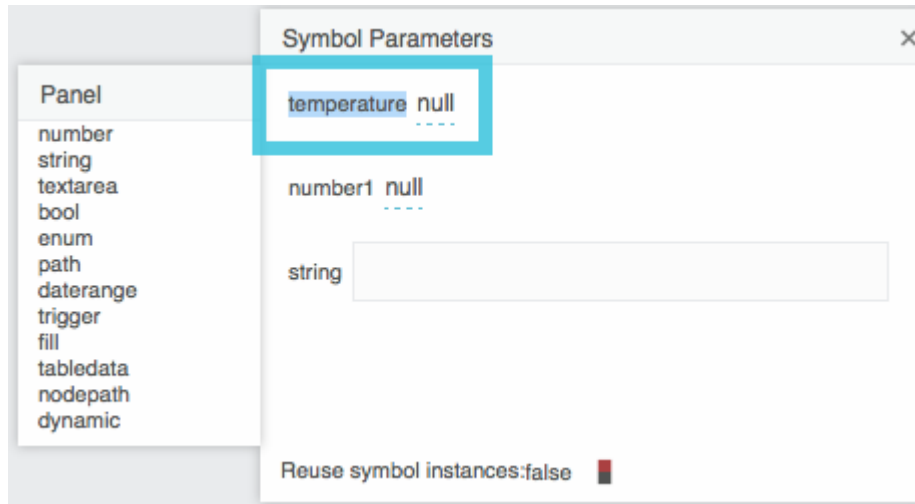


The Symbol Parameters pop-up appears.

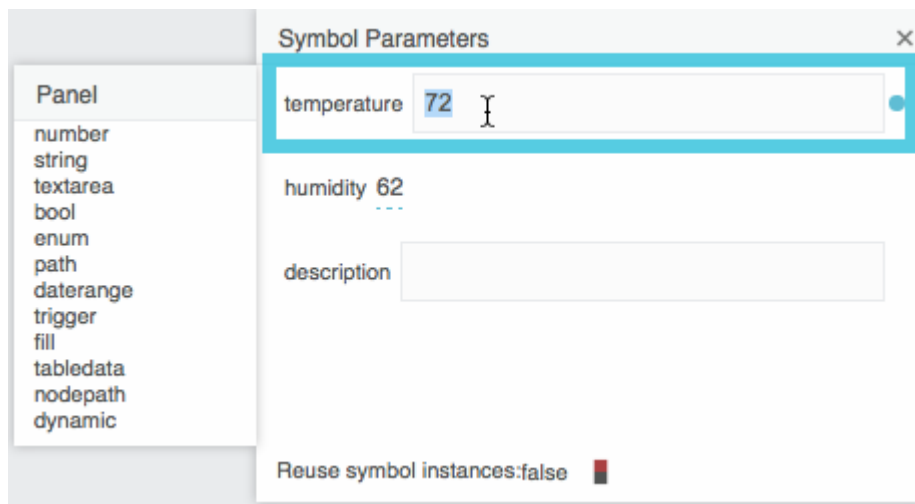
3. Click and drag a data type, and drop the data type on the rightmost portion of the pop-up.



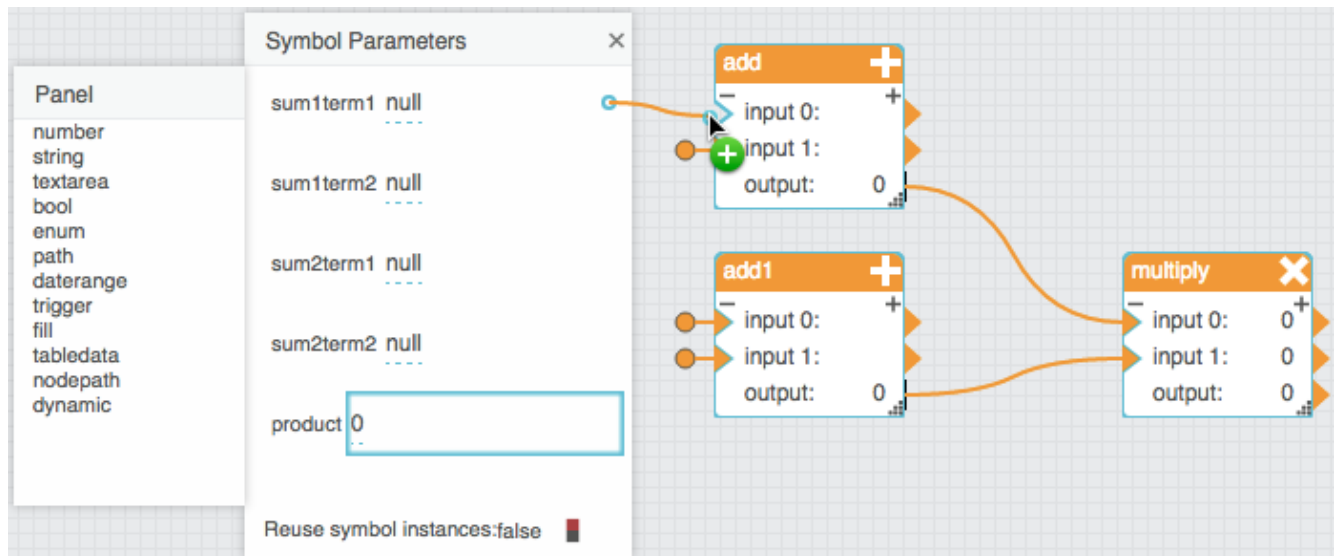
4. If you want to change the label for this parameter, double-click the label and replace it.



5. If you want to define a default value for this parameter, enter the value.



6. If you want to cause this parameter value to determine a block property inside the symbol, create a binding from the parameter to the block property.

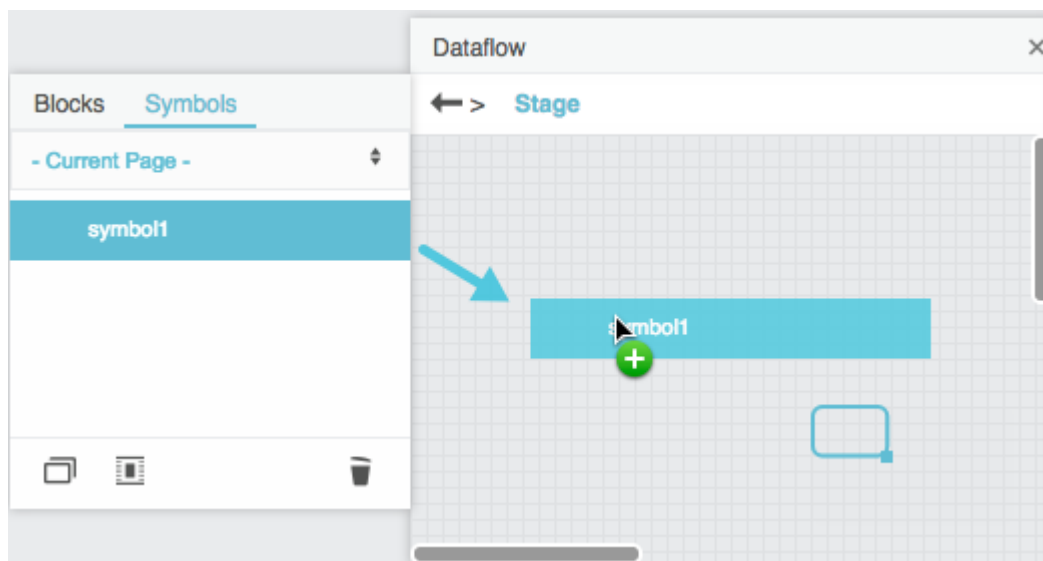


7. Always click **OK** to save changes.

How to Add a Dataflow Symbol Instance

To add a dataflow symbol instance to a dataflow model:

1. Open the dataflow model.
2. Make sure the symbols panel, not the block palette, is selected.
3. Find the symbol name in the symbols panel.
4. Drag the symbol name, and drop it on the dataflow window.



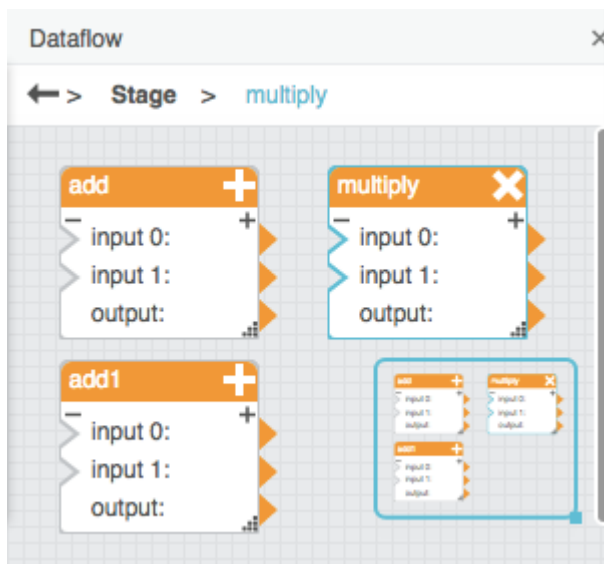
Dataflow Symbol and Repeater Tutorials

These examples show you how to create some simple dataflow symbols and dataflow repeaters.

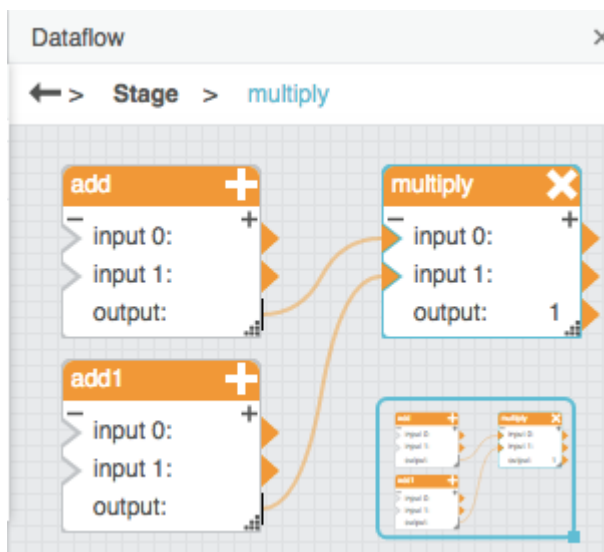
Tutorial: Multiplication Dataflow Symbol

This example shows you how to create a simple dataflow symbol that multiplies two sums.

1. Open any [dataflow model](#).
2. Add a [Multiply](#) block and two [Add](#) blocks to the dataflow model.

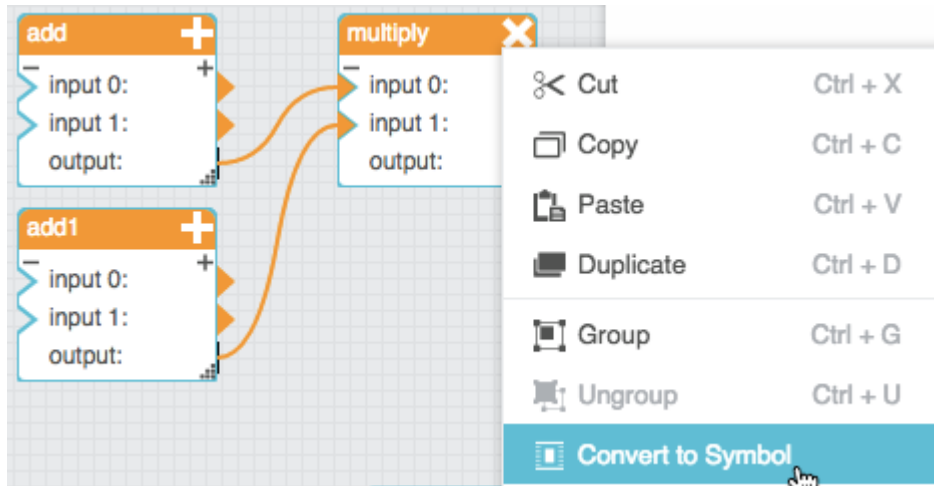


3. Bind the **output** properties of the Add blocks to the **input** properties of the Multiply block.

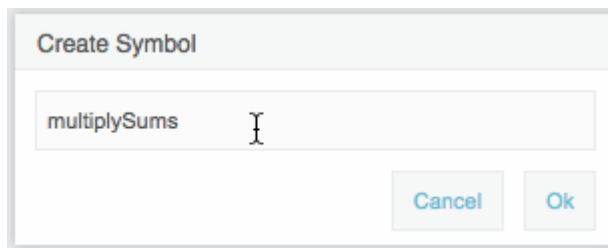


4. Select all three dataflow blocks.


5. Right-click one of the selected blocks, and choose  **Convert to Symbol**.

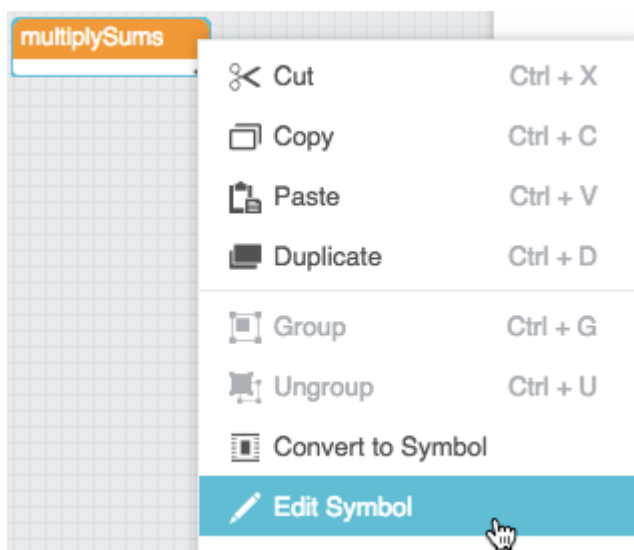


6. Enter multiplySums as the symbol name, and click **OK**.



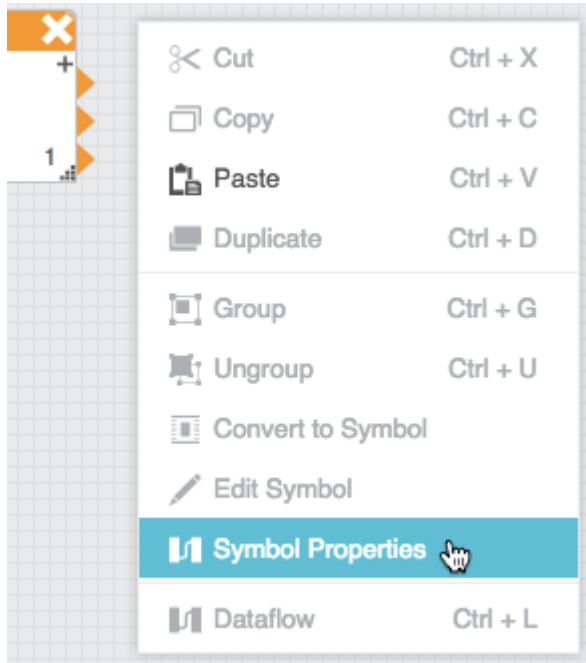
The three blocks are replaced by a symbol instance. In the symbols panel, **multiplySums** appears in the symbols list.

7. Right-click the symbol instance, and select  **Edit Symbol**.

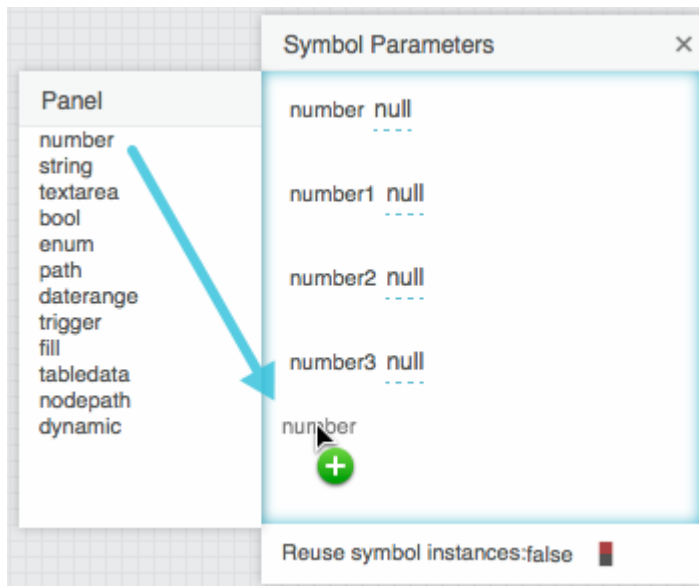


You enter symbol editing mode for the **multiplySums** symbol.

8. Right-click the background of the symbol editing window, and select  **Symbol Properties**.



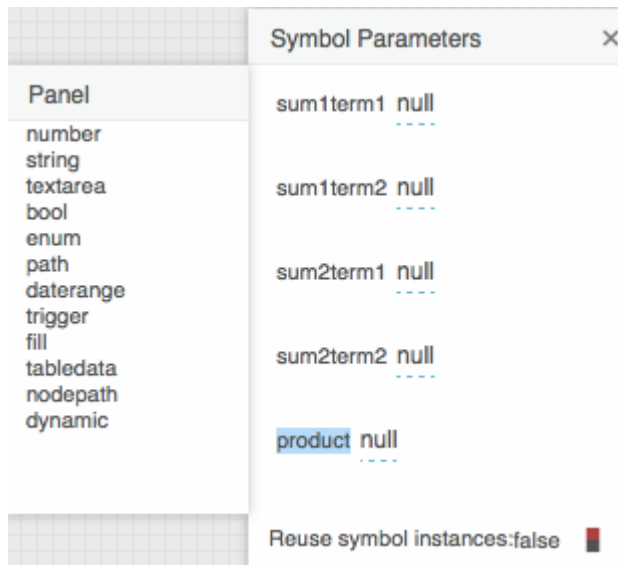
9. Drag five **number** parameters to the rightmost portion of the Symbol Parameters dialog.



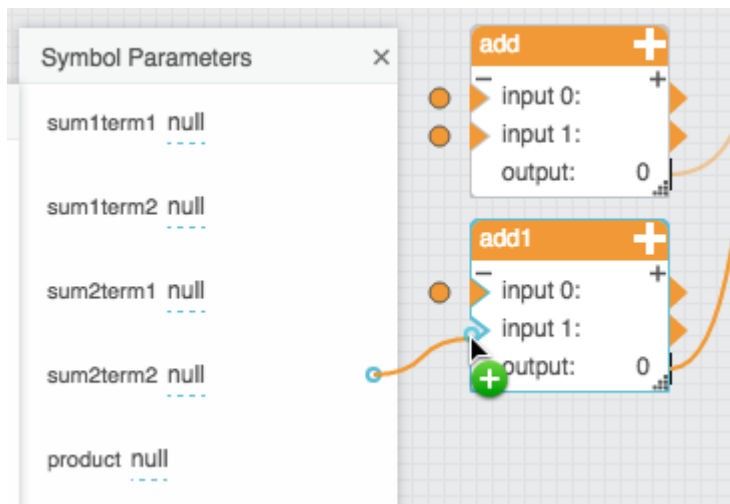
10. Edit the labels of the five parameters to the following labels:

- sum1term1
- sum1term2
- sum2term1
- sum2term2

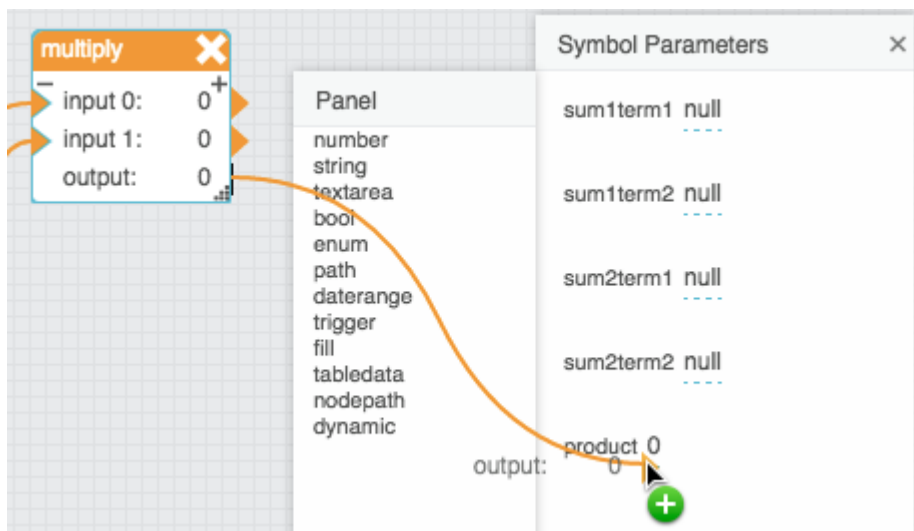
- product



11. Bind the first four symbol parameters to the **input** properties of the Add blocks.



12. Bind the **output** property of the Multiply block to the **product** symbol parameter.



13. Click **OK** to exit symbol editing mode.

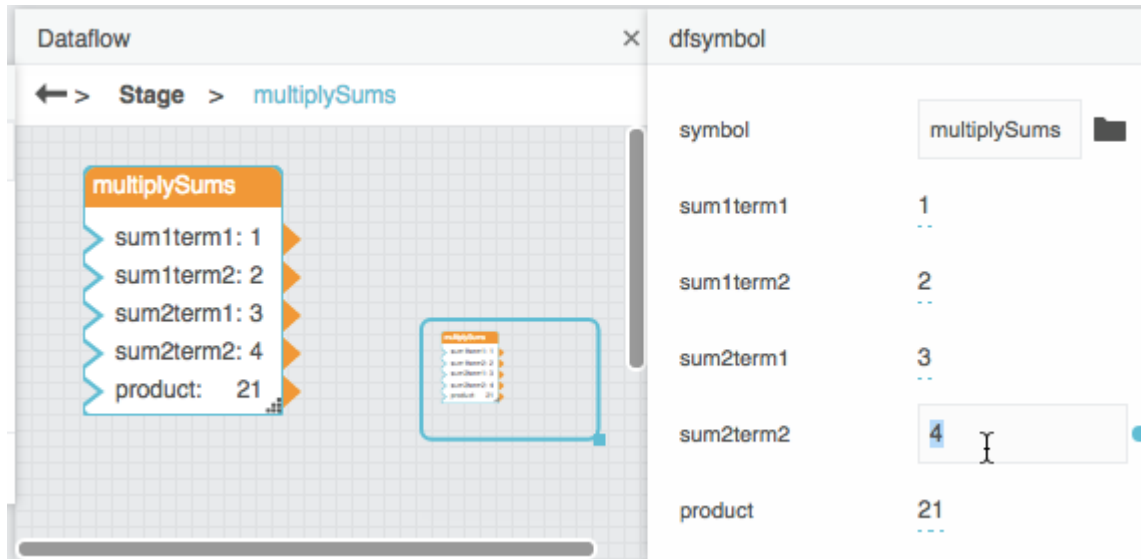
14. In the dataflow window, select the **multiplySums** symbol instance.

The parameters of the **multiplySums** symbol appear in the block properties panel.

15. To cause properties to appear in the visual block in the dataflow window, [pin](#) the properties.

16. To test your symbol, enter numbers for the first four properties.

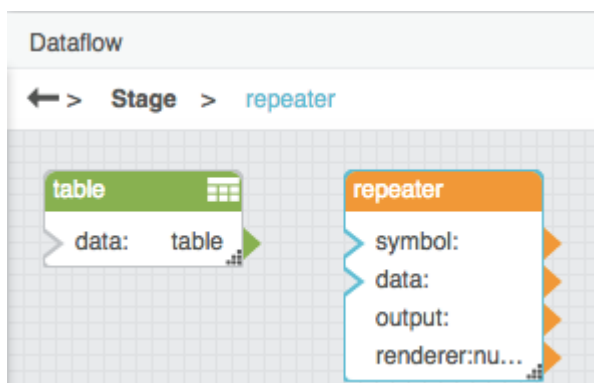
The **product** property updates.



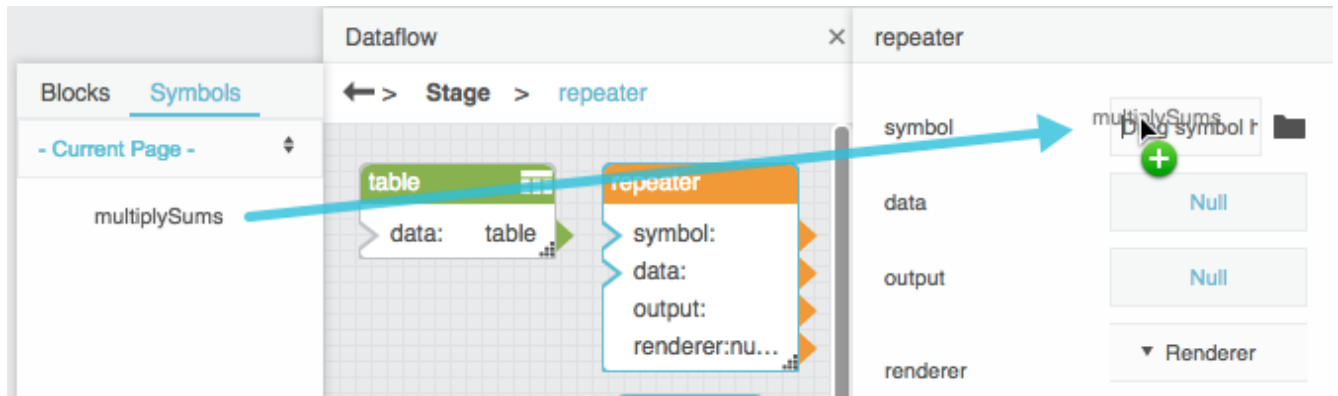
Tutorial: Multiplication Dataflow Repeater

This example uses the dataflow symbol created above to create a simple dataflow repeater.

1. Create the **multiplySums** dataflow symbol as described above.
2. Open any dataflow model.
3. Add a [Table](#) block and a Repeater block to the dataflow model.



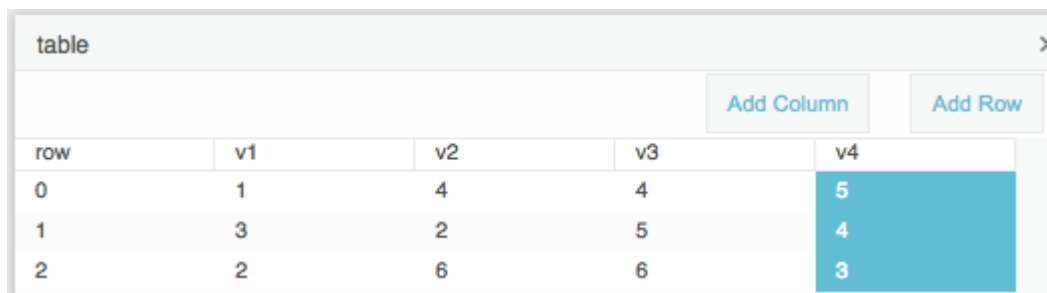
- 4. With the Repeater block selected, open the symbols panel, and drag the **multiplySums** symbol name to the **symbol** property.



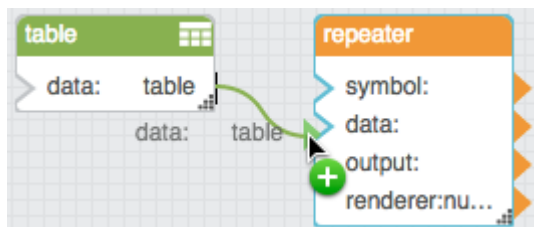
- 5. Select the Table block, and invoke the value of the **data** property to open the table.



- 6. [Edit the table](#) so that the table contains four columns of numbers.



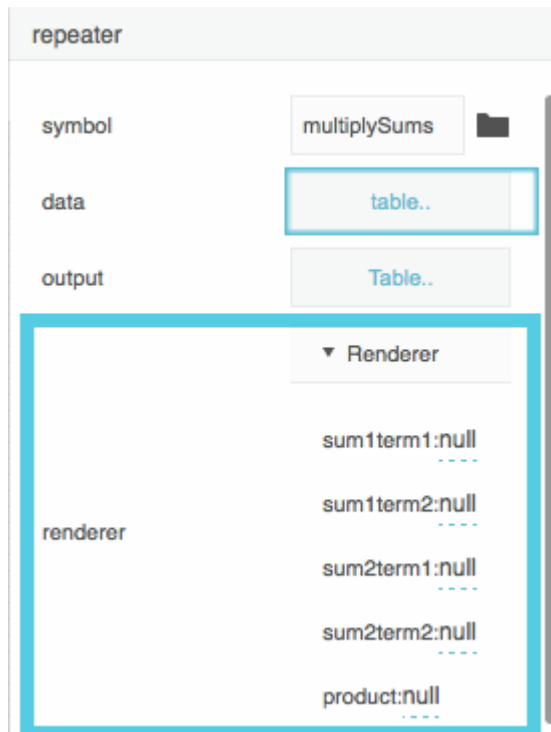
- 7. [Bind](#) the **data** property of the Table block to the **data** property of the Repeater block.



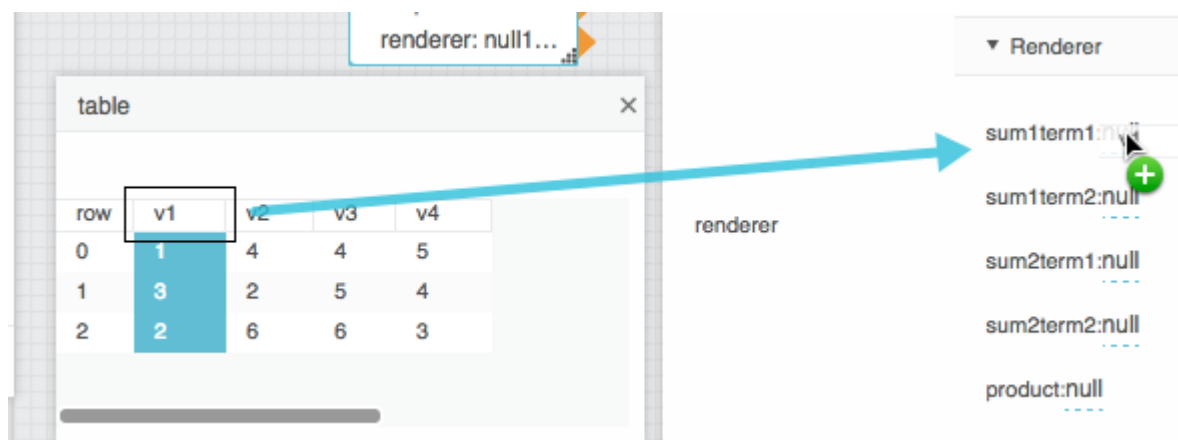
- 8. With the Repeater block selected, invoke the value of the **data** property to open the table.



9. Expand the **Renderer** section in the block properties panel.



10. Bind the table columns to the first four **Renderer** properties by dragging each column header.



11. To view the output of the repeater, invoke the value of the **output** property.



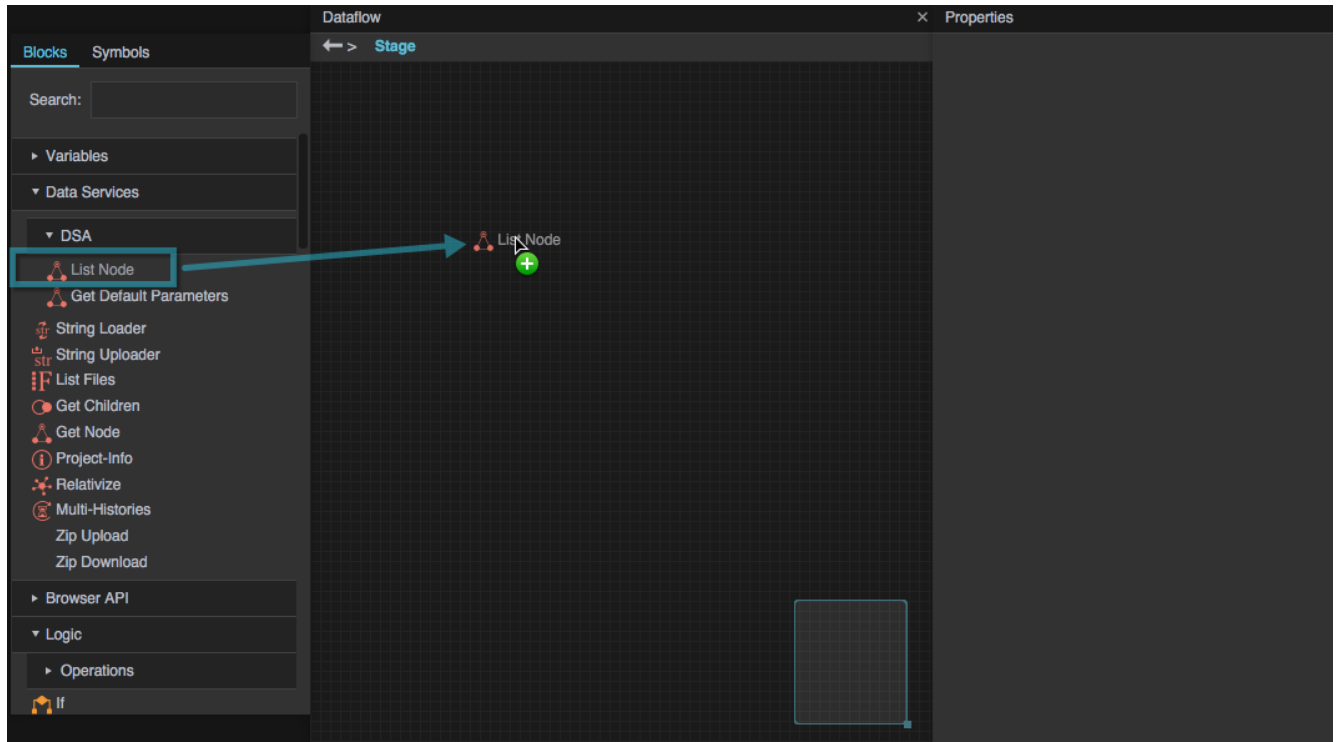
This output table contains a row for each input table row. It contains a column for each symbol property, including the product property.

row	sum1term1	sum1term2	sum2term1	sum2term2	product
0	1	4	4	5	45
1	3	2	5	4	45
2	2	6	6	3	72

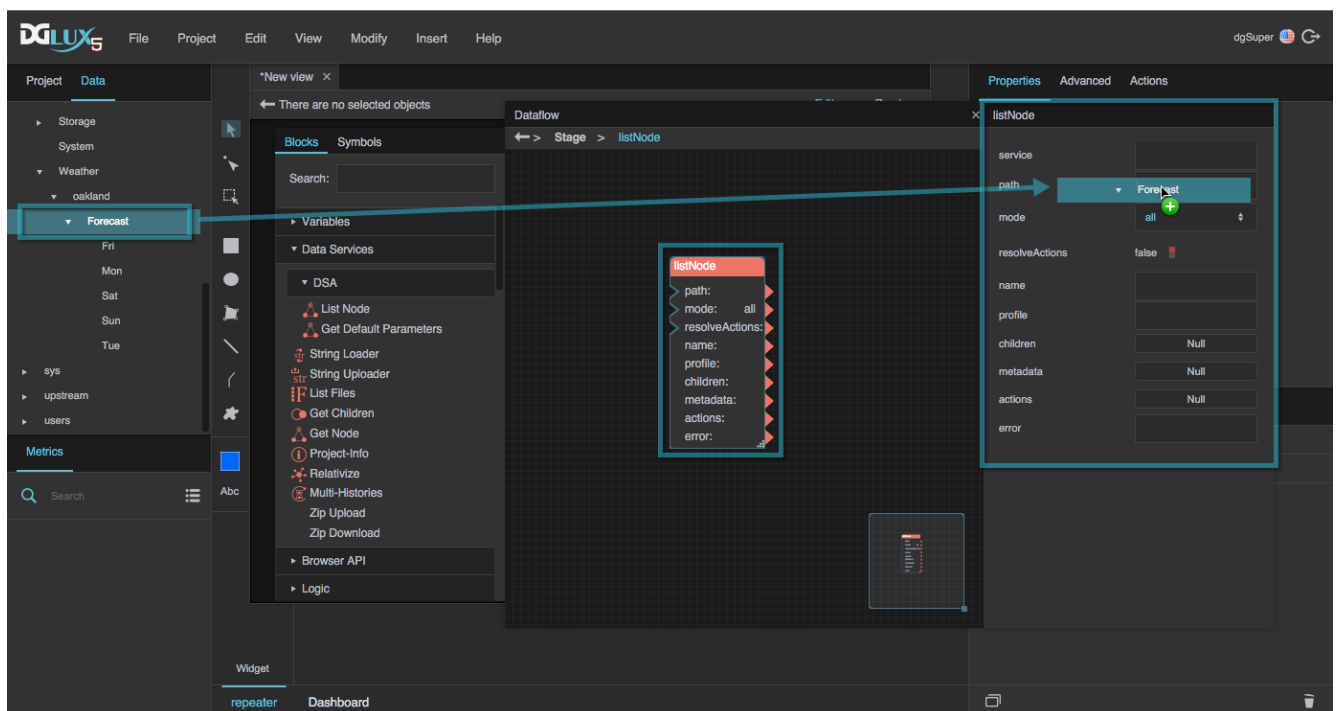
Tutorial: City Weather Tracking Dataflow Symbol

This example shows you how to create a dataflow symbol that combines data collected by [Load Value](#) blocks. This particular example requires the [DSA Weather DSLink](#). To connect to the Weather DSLink, follow the steps in [Access DSLink data](#).

1. Open any dataflow model.
2. Add a List Node block.



3. In the **Data panel**, find **downstream > Weather > <city> > Forecast**. With the List Node block selected, drag the **Forecast** node to the **path** property.



4. Invoke the value of the List Node block's **children** property to view the table of children.

The screenshot shows the DG LUX 5 IDE interface. The main window displays a Dataflow graph with a single node named 'listNode'. The node's properties are visible in the right-hand pane, including 'path', 'mode', 'resolveActions', 'name', 'profile', 'children', 'metadata', 'actions', and 'error'. The 'children' property is highlighted, and a 'Table...' button is visible next to it. Below the main window, a 'Table' widget is open, displaying a table with the following data:

row	name	path	base	profile	type	invokable	writable
0	Fri	/downstream/Weather/Oakland-CA-United States/Forecast/Fri	/downstream/Weather	node			
1	Mon	/downstream/Weather/Oakland-CA-United States/Forecast/Mon	/downstream/Weather	node			
2	Sat	/downstream/Weather/Oakland-CA-United States/Forecast/Sat	/downstream/Weather	node			
3	Sun	/downstream/Weather/Oakland-CA-United States/Forecast/Sun	/downstream/Weather	node			
4	Tue	/downstream/Weather/Oakland-CA-United States/Forecast/Tue	/downstream/Weather	node			

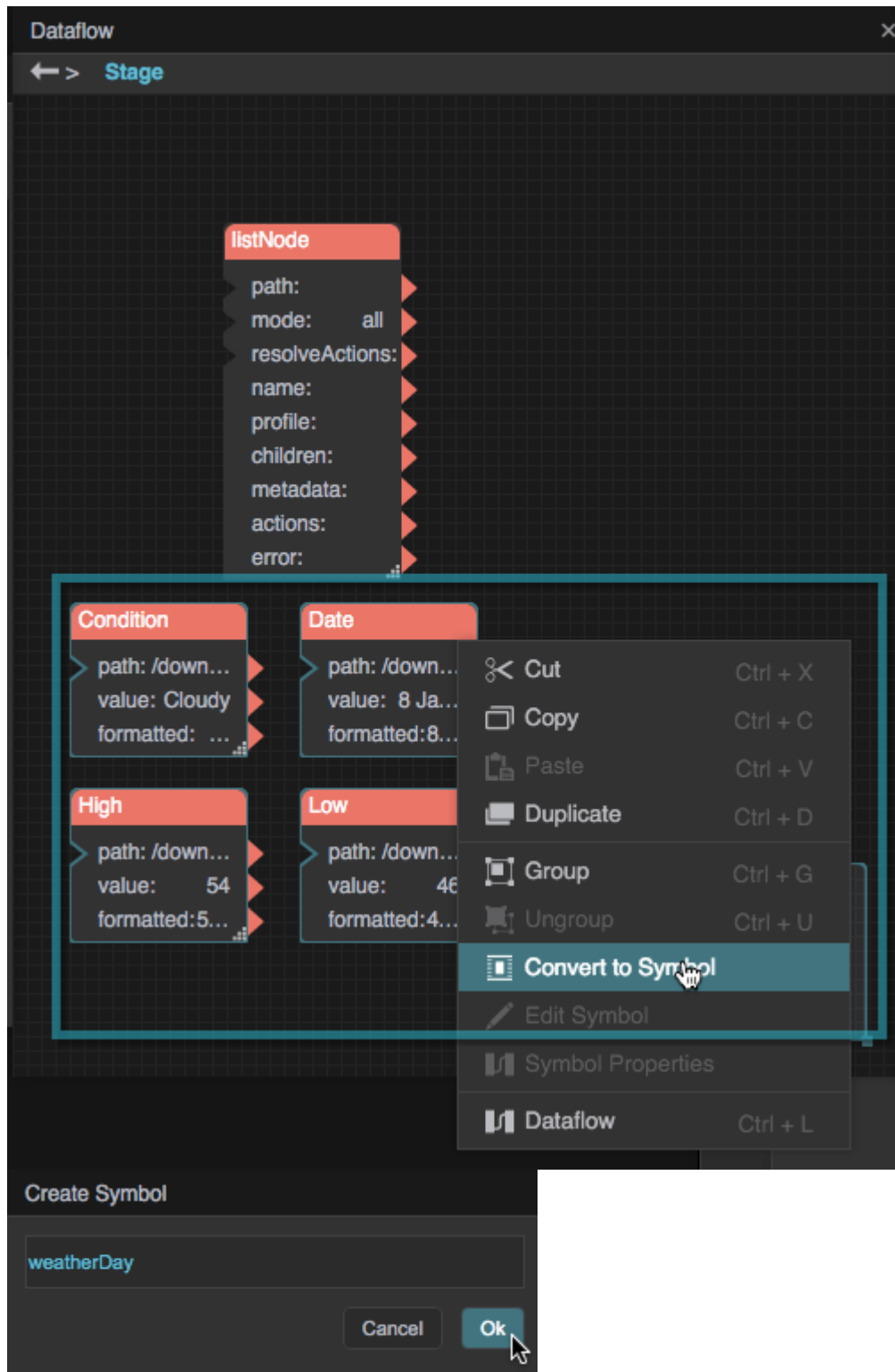
5. In the Data panel, select one of the children of the **Forecast** node.

The screenshot shows the 'Data' panel in the DG LUX 5 IDE. The 'Forecast' node is selected, and its children are listed: Fri, Mon, Sat, Sun, and Tue. The 'Fri' node is highlighted.

6. Drag each of this node's metrics from the **Metrics panel** to the dataflow window.

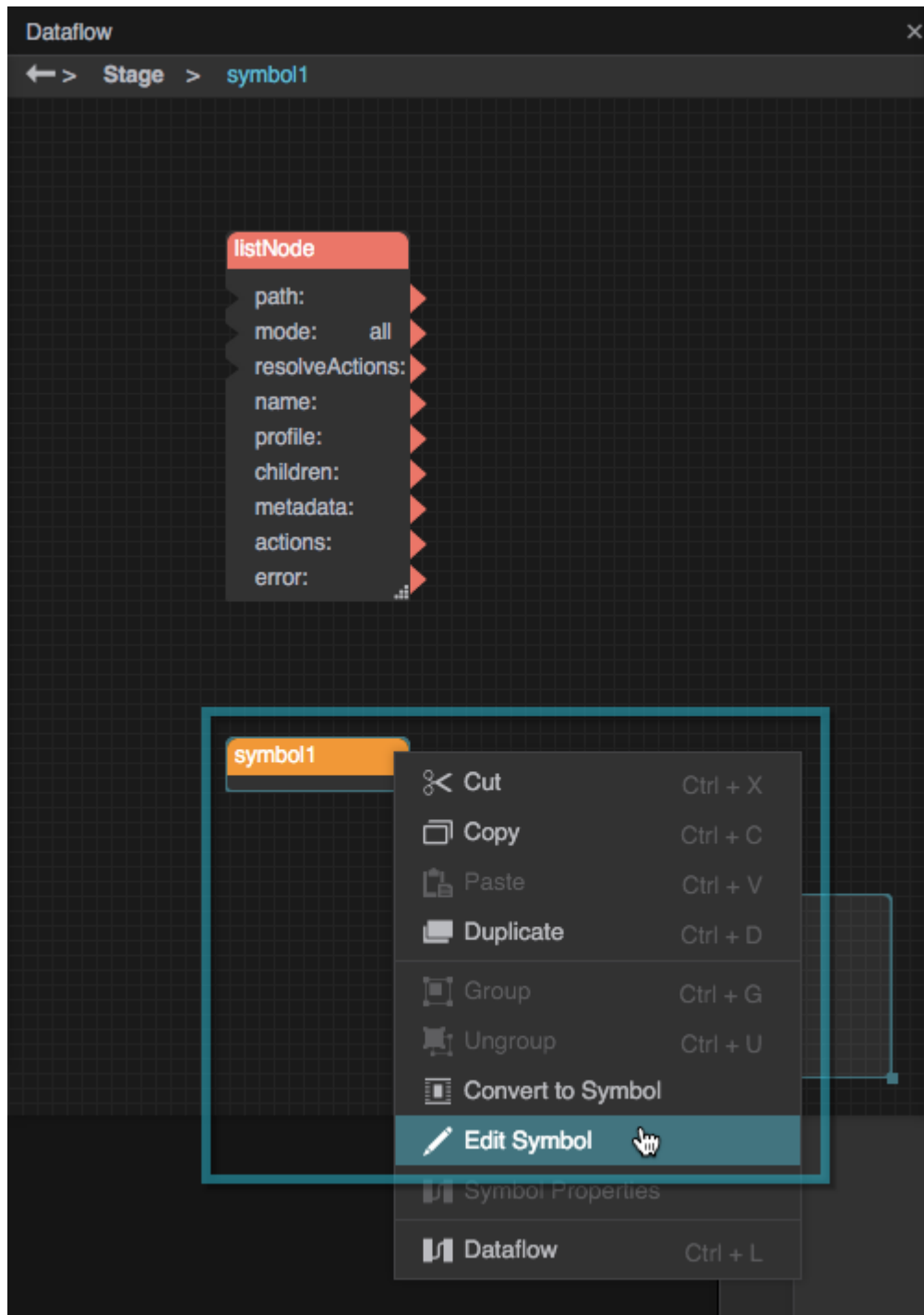
The screenshot displays a dataflow editor with a dark theme. On the left, a 'Project Data' sidebar shows a tree view with folders like 'haystack', 'mqtt', 'relayr', 'REST', 'Slack', 'splunk', 'Storage', 'System', 'Weather', and 'oakland'. Under 'Weather', there is a 'Forecast' folder containing a 'Fri' block, which is highlighted. Below the sidebar is a 'Metrics' section with a search bar and a list of metrics: 'Condition : Cloudy', 'Date : 8 Jan 2016', 'High : 54 °F', and 'Low : 46 °F'. The 'Low : 46 °F' metric is highlighted with a red box, and a red arrow points from it to a 'Low: 46 °F' block in the main workspace. The main workspace shows a 'Stage' view with a 'High' sub-stage. Inside the 'High' sub-stage, there are four 'Load Value' blocks: 'Condition', 'Date', 'High', and 'Low: 46 °F'. The 'Condition' block has a value of 'Cloudy', the 'Date' block has a value of '8 Jan 2016', and the 'High' block has a value of '54'. The 'Low: 46 °F' block is highlighted with a red box. A 'Properties' panel on the right shows the details of the selected 'Low: 46 °F' block, including fields for 'path', 'mode', 'resolveActions', 'name', 'profile', 'children', 'metadata', 'actions', and 'error'. The 'List Node' block is also visible in the workspace, with its properties shown in a separate panel.

7. Select the four Load Value blocks and select  **Convert to Symbol**. Name the symbol weatherDay.



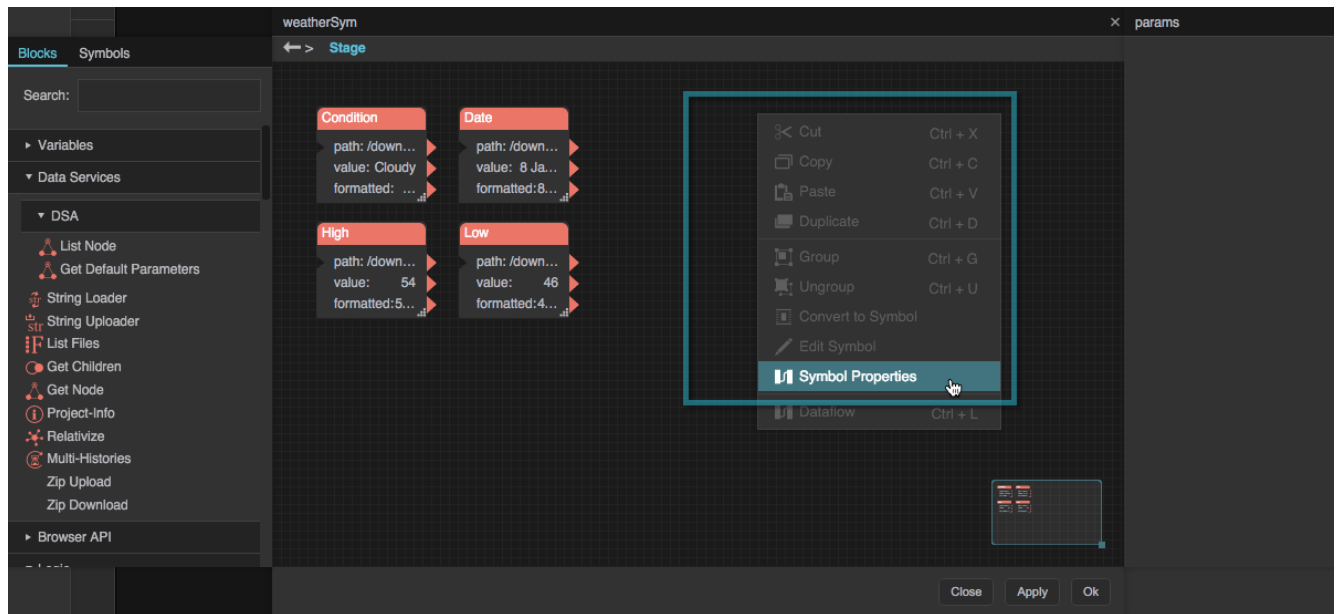
The four blocks are replaced by a single symbol block.

8. Right-click the symbol block and select  **Edit Symbol**.

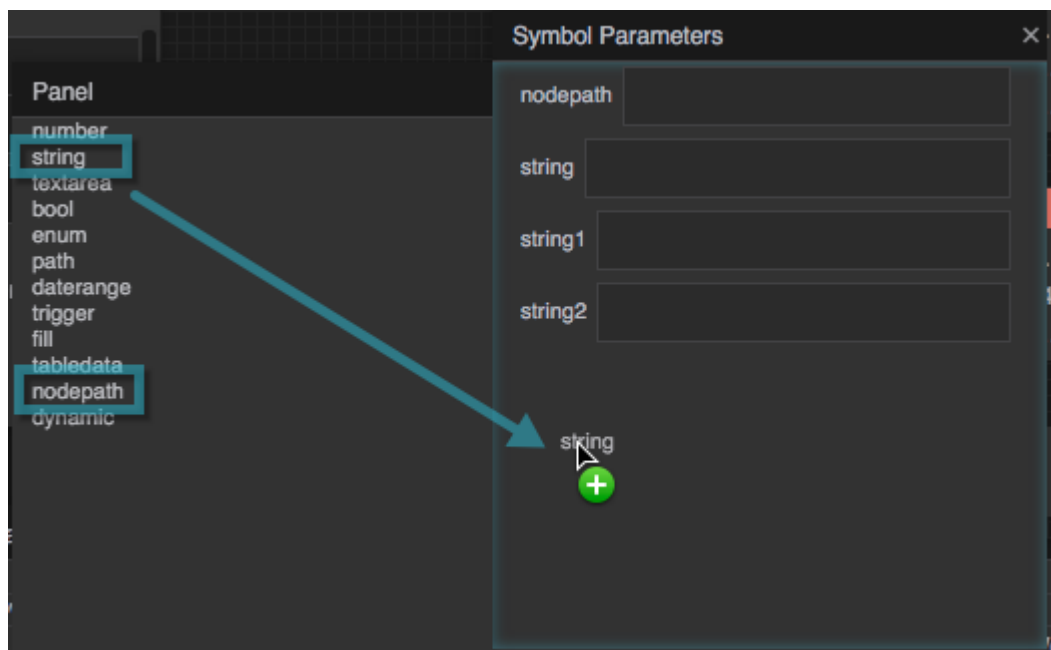


You enter symbol editing mode for the **weatherDay** symbol.

9. Right-click the background of the symbol editing window, and select **Symbol Properties**.

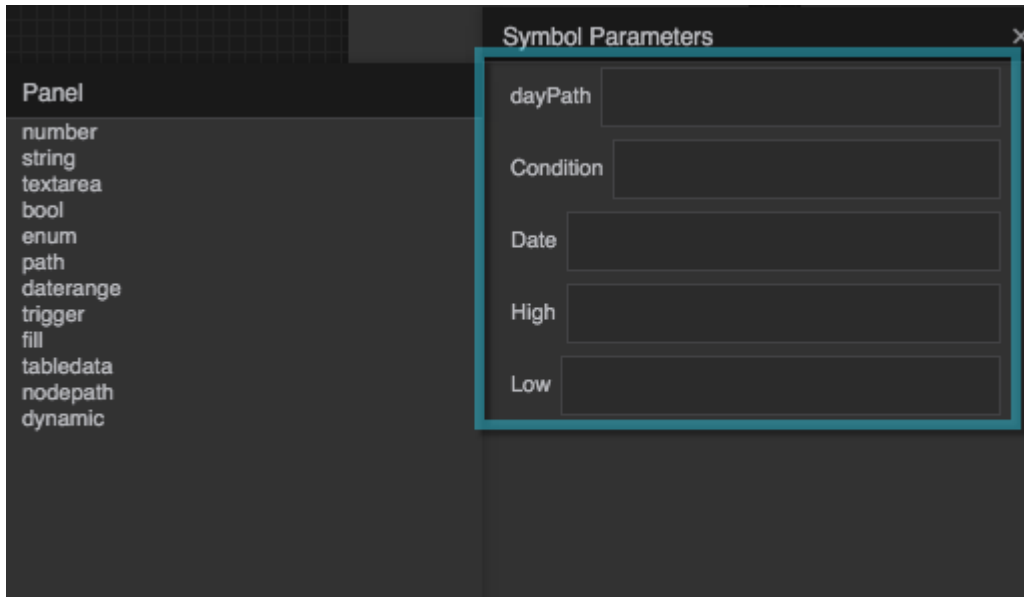


10. Add a **nodepath** property and four **string** properties.

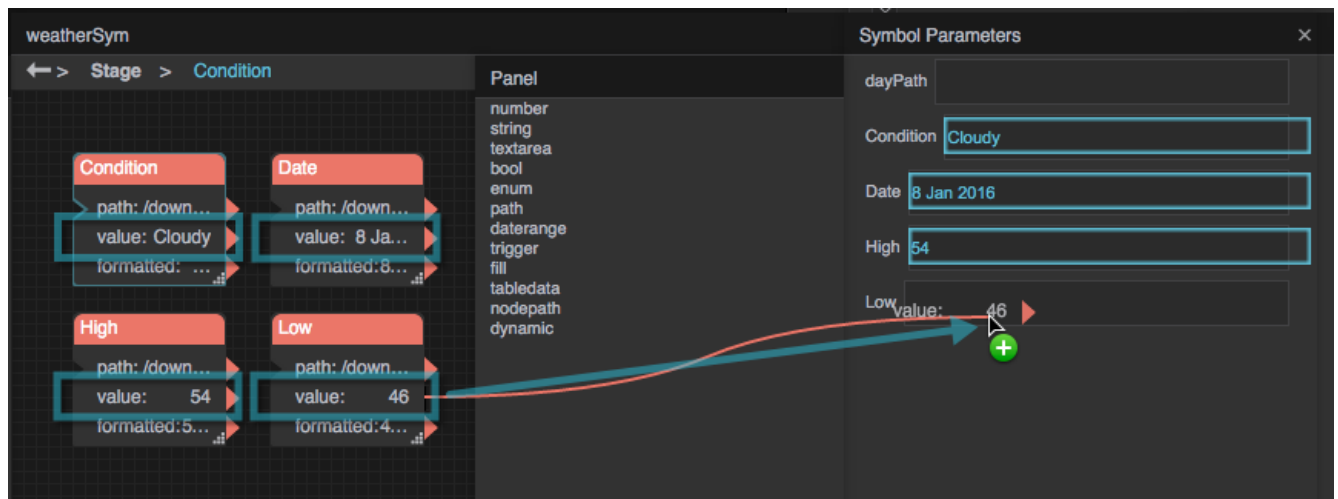


11. Name the properties as follows:

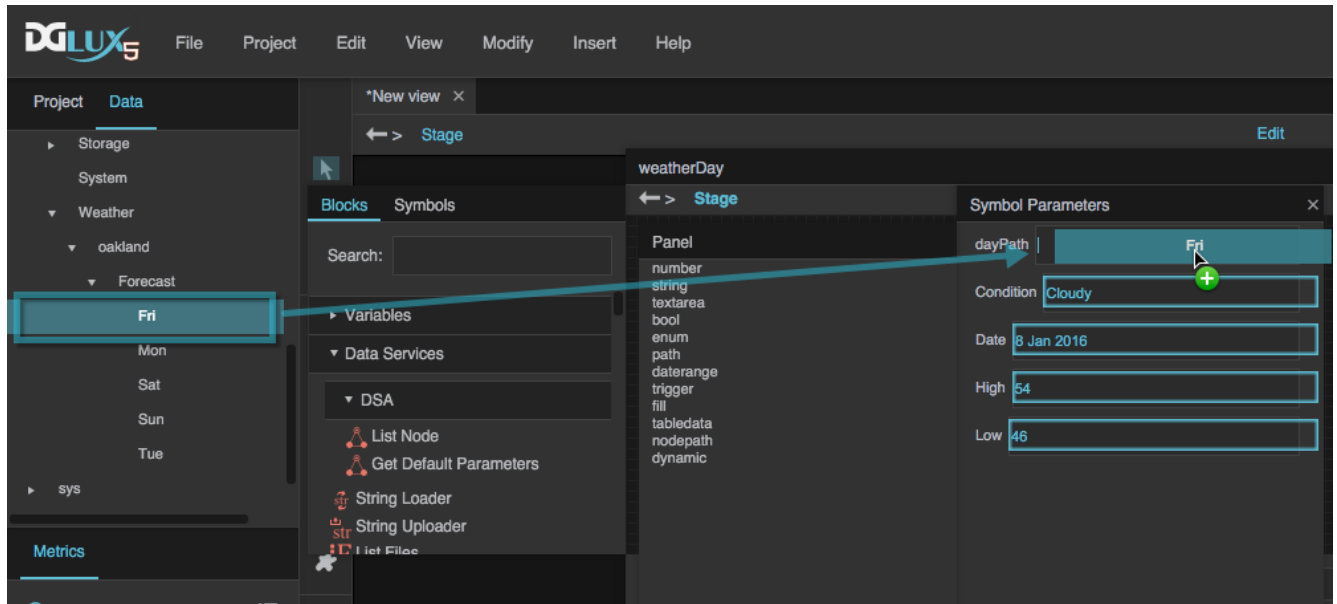
- dayPath
- Condition
- Date
- High
- Low



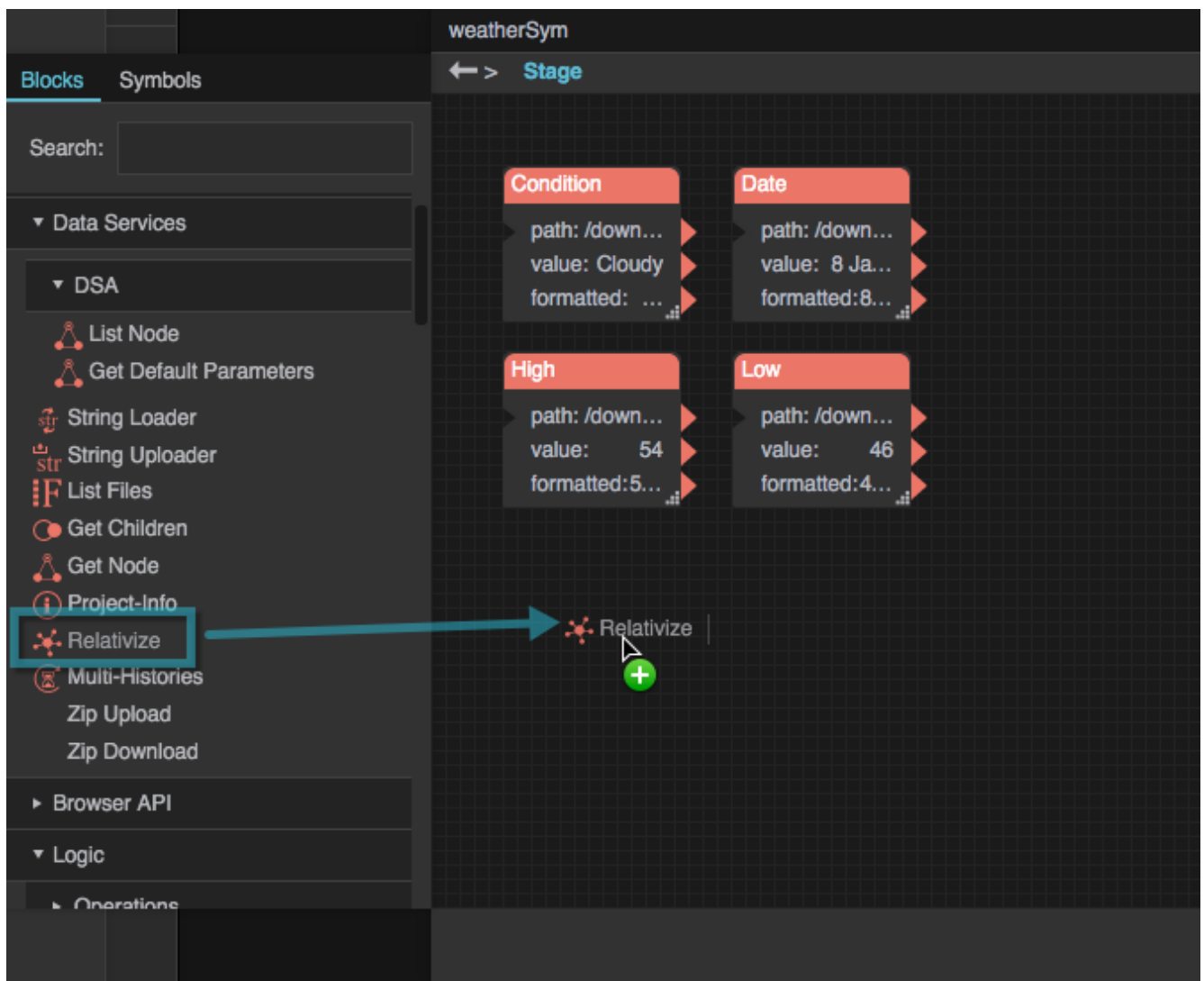
12. Bind the **value** properties of the List Node blocks to the **string** symbol parameters.



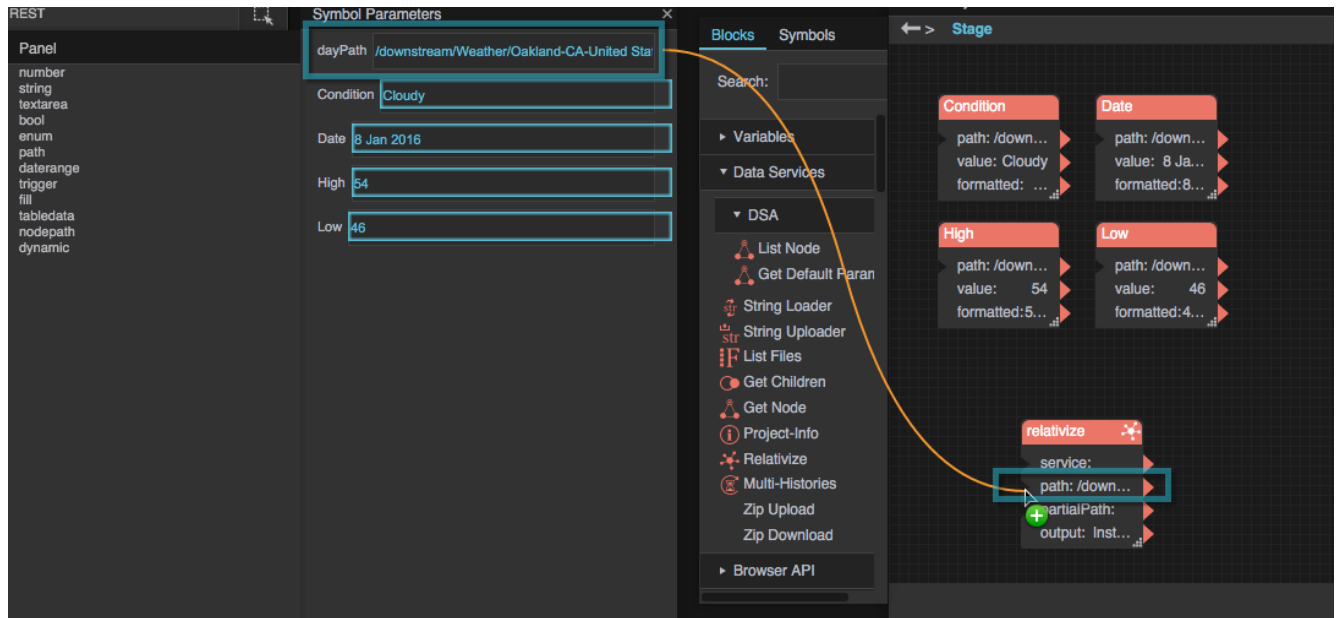
13. Bind one of the children of the **Forecast** node from the Data panel to the dayPath symbol parameter.



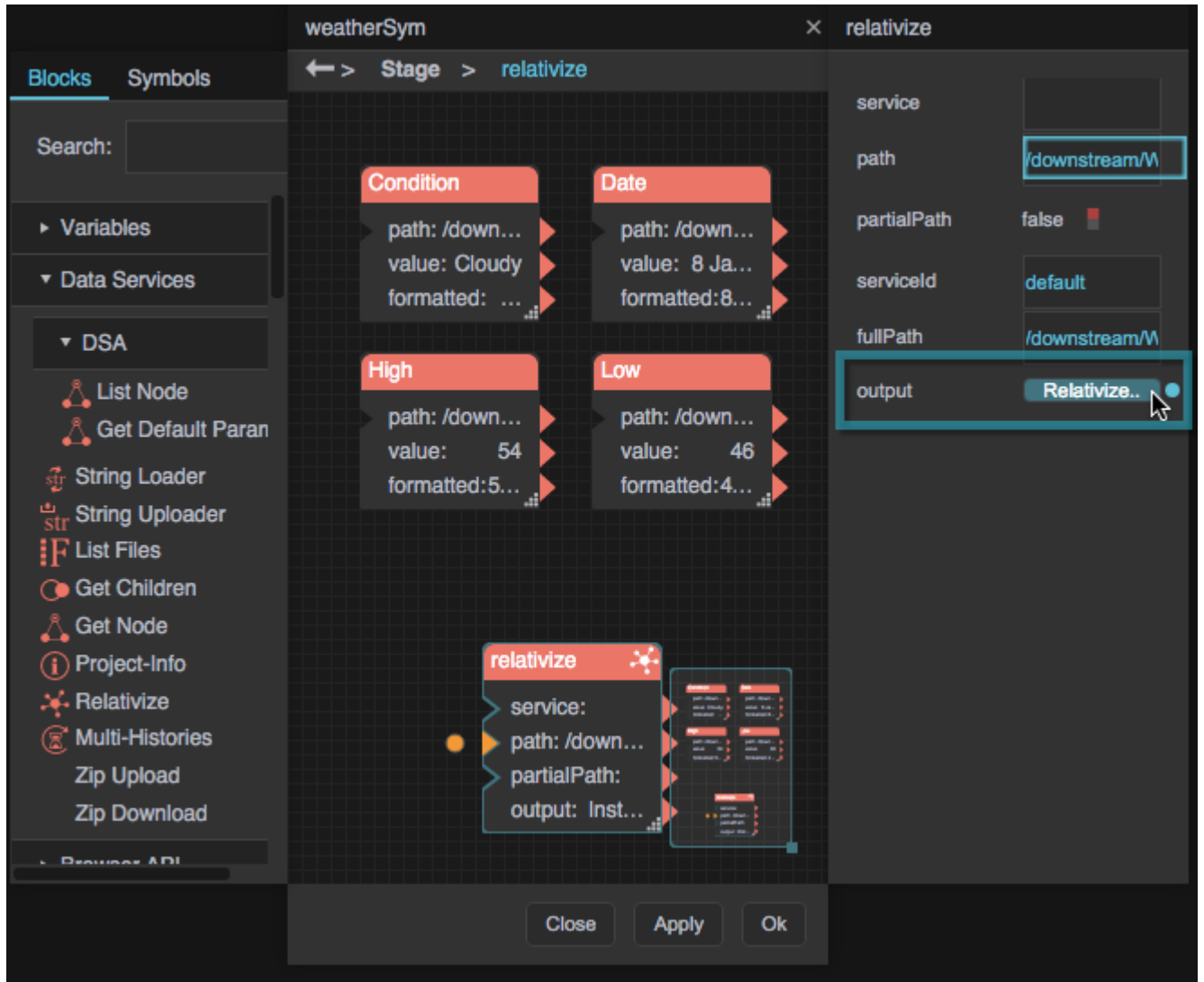
14. Add a **Relativize** block to the symbol editing window.



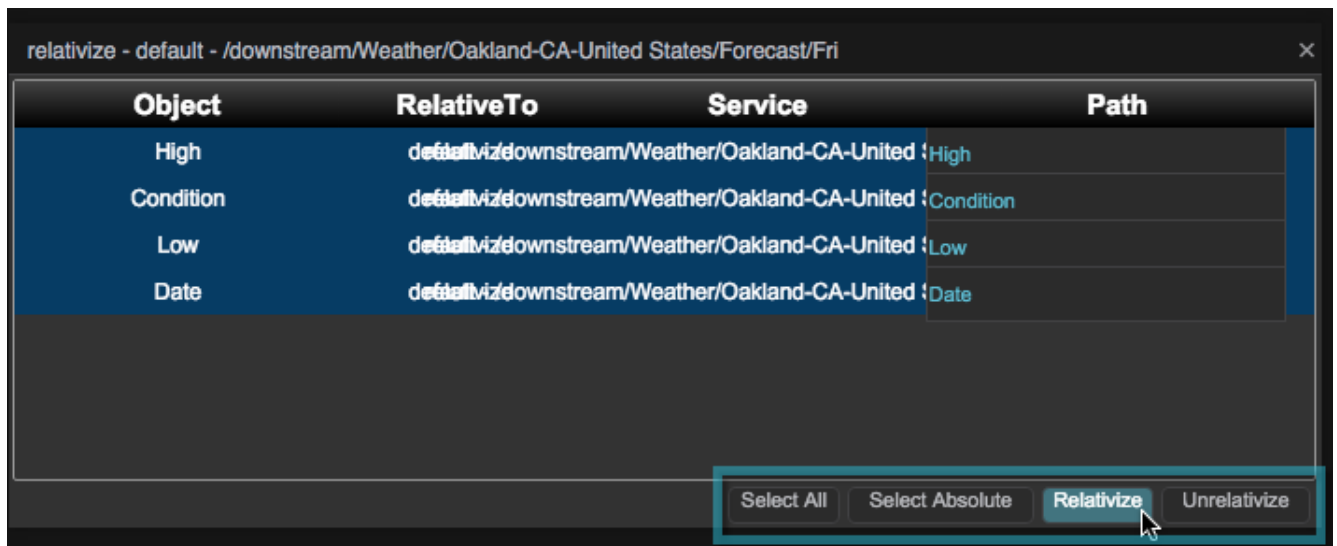
15. Bind the dayPath symbol parameter to the **path** property of the Relativize block.



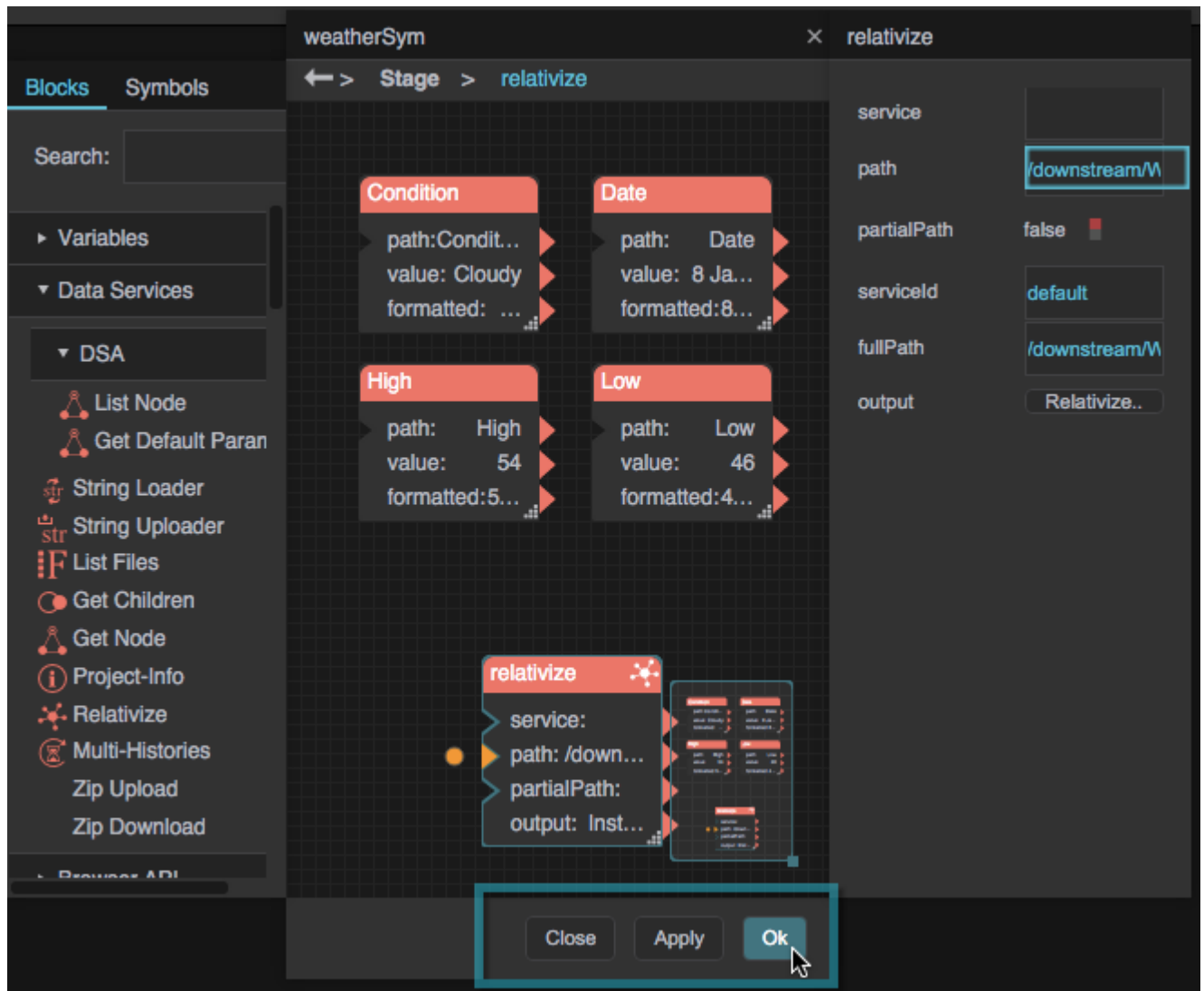
16. Select the Relativize block, and click the **Relativize** button next to **output**.



17. In the pop-up, click **Select All**, and then click **Relativize**.



18. Close the pop-up, and click **OK** to close the symbol editing window.



19. To test your symbol, select the symbol and change the value of the **dayPath** symbol property. The other four properties update when you do this.

symbol	weatherSym
dayPath	/downstream/Weather/Oakland-CA-United States/Forecast/
Condition	Cloudy
Date	8 Jan 2016
High	54
Low	46

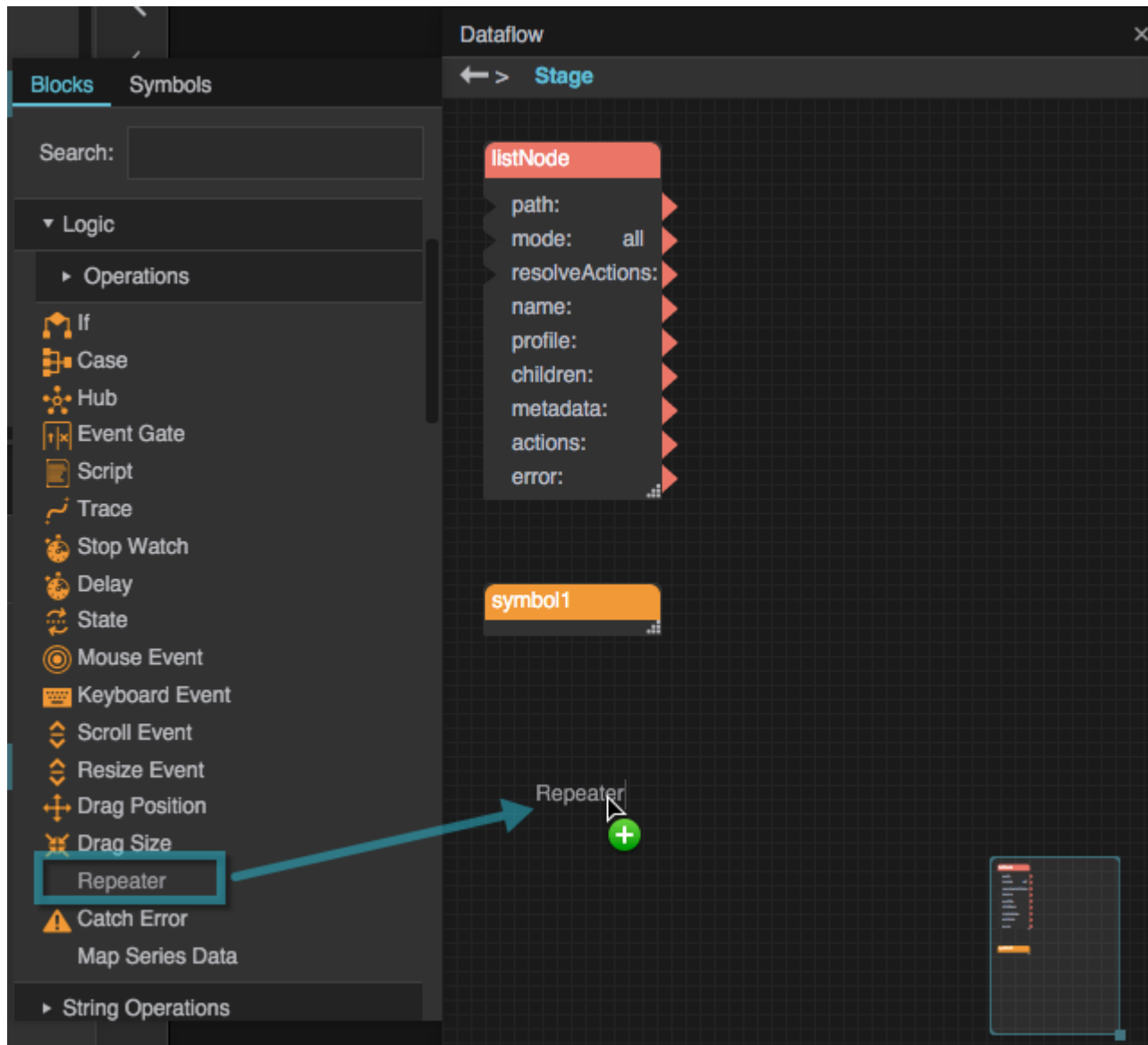
If you want, you can [pin](#) properties so that they will visually appear in the block.

Tutorial: City Weather Tracking Dataflow Repeater

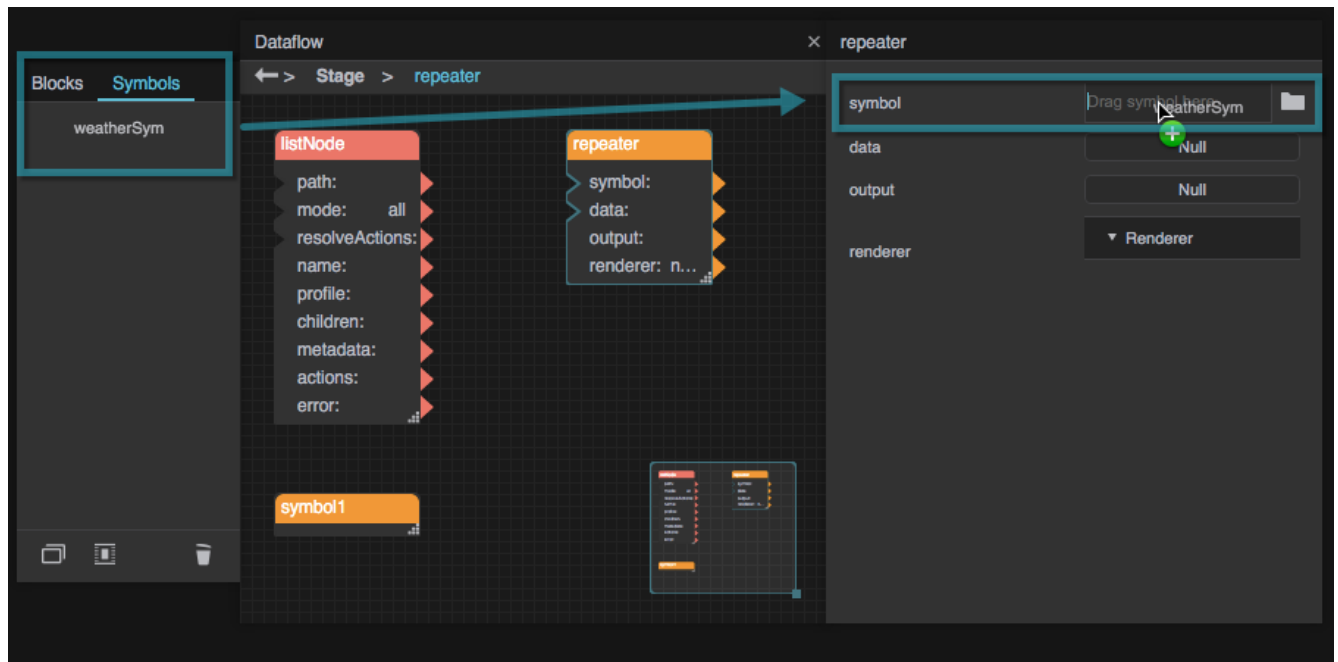
This example uses the `weatherDay` symbol to create a dataflow repeater.

If you used non-DSA data to create the symbol, use the **output** property of the [Get Children](#) block instead of the **children** property of the [List Node](#) block.

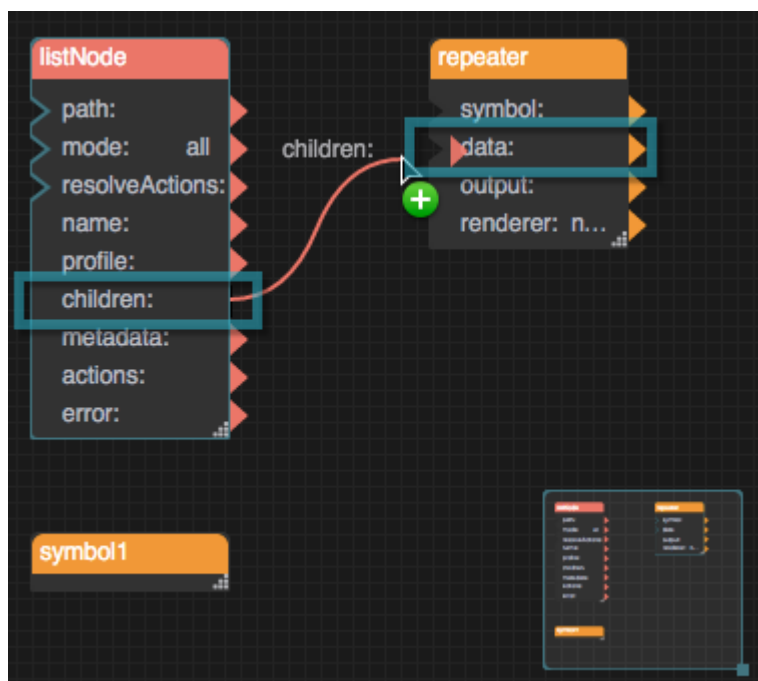
1. Create the `weatherDay` dataflow symbol as described above.
2. Add a Repeater block to the dataflow window.



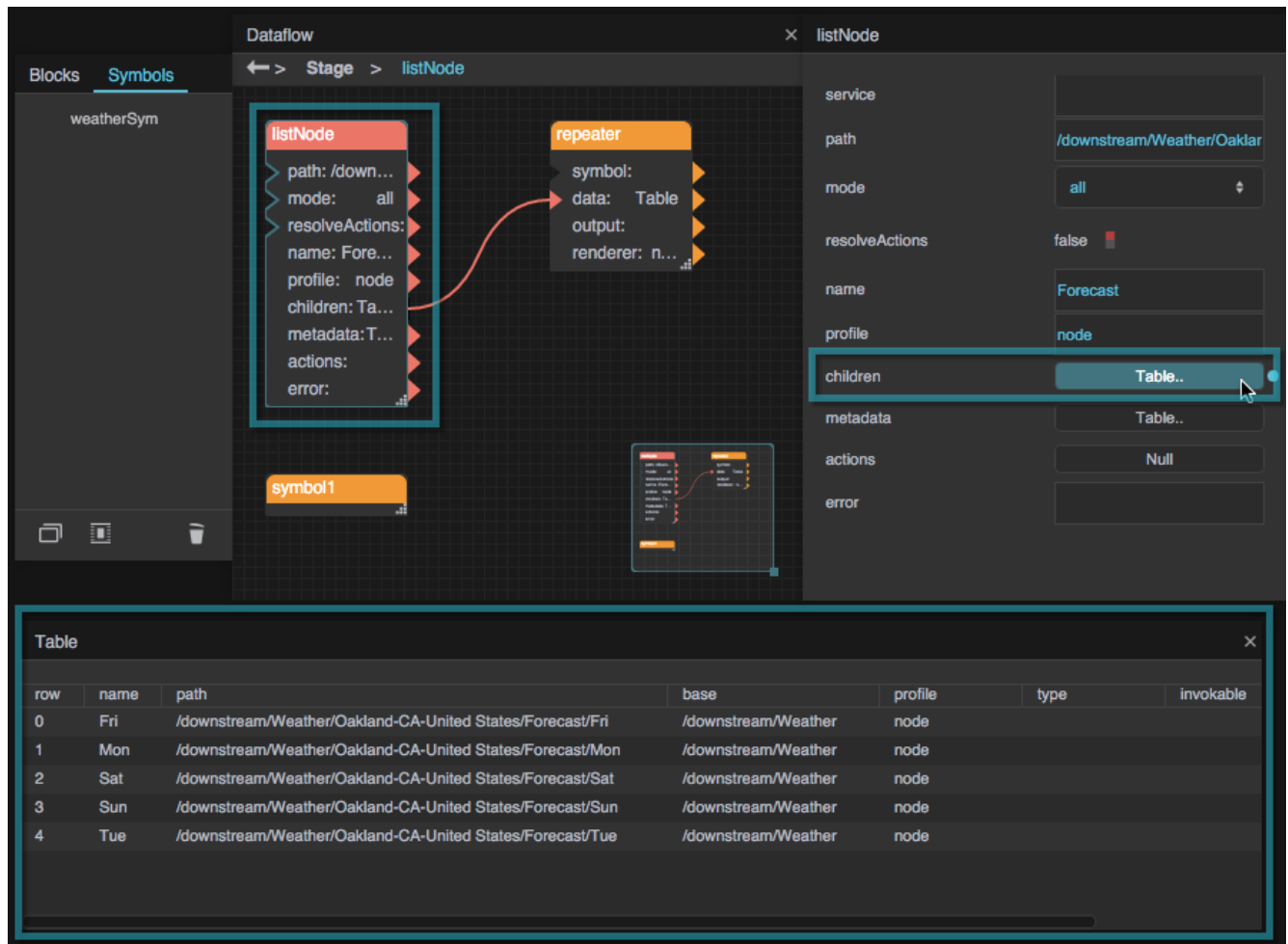
3. With the Repeater block selected, open the **Symbols** panel of the dataflow window, and drag **weatherDay** to the **symbol** property.



4. Select the **List Node** block that you created when you made the weatherDay symbol, and bind the **children** table to the **data** property of the repeater.



5. With the List Node block selected, click the button next to **children** to open the table.



The screenshot shows the Dataflow IDE interface. On the left, there's a 'weatherSym' symbol. In the center, a 'listNode' block is connected to a 'repeater' block. The 'listNode' block has the following properties: path: /down..., mode: all, resolveActions: (empty), name: Fore..., profile: node, children: Ta..., metadata: T..., actions: (empty), error: (empty). The 'repeater' block has: symbol: (empty), data: Table, output: (empty), renderer: n... . A 'Table' block is selected in the 'children' property of the 'repeater' block. A 'Table' dialog is open, showing a table with columns: row, name, path, base, profile, type, and invocable. The table contains 5 rows of data representing days of the week and their corresponding paths and profiles.

row	name	path	base	profile	type	invokable
0	Fri	/downstream/Weather/Oakland-CA-United States/Forecast/Fri	/downstream/Weather	node		
1	Mon	/downstream/Weather/Oakland-CA-United States/Forecast/Mon	/downstream/Weather	node		
2	Sat	/downstream/Weather/Oakland-CA-United States/Forecast/Sat	/downstream/Weather	node		
3	Sun	/downstream/Weather/Oakland-CA-United States/Forecast/Sun	/downstream/Weather	node		
4	Tue	/downstream/Weather/Oakland-CA-United States/Forecast/Tue	/downstream/Weather	node		

- Select the Repeater block, and bind the **path** table column to the **dayPath** renderer property by dragging the column header.

The screenshot shows the Dataflow IDE interface. On the left, a 'listNode' is connected to a 'repeater' node. The 'repeater' node's configuration panel on the right shows the 'output' field set to 'Table..'. Below this, the 'dayPath' field is set to 'path', and a '+' button is visible next to it. A blue arrow points from this '+' button to a 'Table' window at the bottom of the screen. The 'Table' window displays the following data:

row	name	path	base	profile	type	invokable
0	Fri	/downstream/Weather/Oakland-CA-United States/Forecast/Fri	/downstream/Weather	node		
1	Mon	/downstream/Weather/Oakland-CA-United States/Forecast/Mon	/downstream/Weather	node		
2	Sat	/downstream/Weather/Oakland-CA-United States/Forecast/Sat	/downstream/Weather	node		
3	Sun	/downstream/Weather/Oakland-CA-United States/Forecast/Sun	/downstream/Weather	node		
4	Tue	/downstream/Weather/Oakland-CA-United States/Forecast/Tue	/downstream/Weather	node		

7. To view the output of the repeater, click the button next to **output**.

The screenshot shows a dataflow editor interface. On the left, a 'listNode' block is connected to a 'repeater' block. The 'repeater' block's configuration is shown on the right, with the following settings:

- symbol: weatherSym
- data: Table..
- output: Table..
- renderer: (expanded)

 - dayPath: [empty field]
 - Condition: [empty field]
 - Date: [empty field]
 - High: [empty field]
 - Low: [empty field]

Below the editor, a table window displays the output data:

row	dayPath	Condition	Date	High	Low
0	/downstream/...	Cloudy	8 Jan 2016	54	46
1	/downstream/...	PM Showers	11 Jan 2016	61	44
2	/downstream/...	AM Showers	9 Jan 2016	58	44
3	/downstream/...	Mostly Sunny	10 Jan 2016	61	47
4	/downstream/...	Mostly Sunny	12 Jan 2016	62	47

This output table contains a row for each instance of the dataflow symbol.

[Previous: Dataflow](#)

[Next: Working with Tables](#)

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

Permanent link: https://wiki.dglogik.com/dglux5_wiki:dataflow:symbols_repeater:home

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

