Repeater

Block Group:	Logic
Icon:	no icon

The Repeater block returns a table that, for each row in an input table, lists the parameter values of an input dataflow symbol.

For information on using dataflow blocks, see Dataflow.

For information on creating dataflow repeaters, see Dataflow Symbols and Dataflow Repeaters.

Input/Output Properties

The following properties of the Repeater block can take input and give output.

- symbol *(name of a dataflow symbol)*
- data *(table)*
- mode *(enum)*
- thread *(number)*
- streamQueue *(number)*
- keyColumn *(string)*
- renderer properties

thread, **streamQueue** and **keyColumn** are not shown on the block; edit them in the Properties panel.

symbol specifies the dataflow symbol to repeat.

data specifies the table that determines the number of symbol instances and the properties of each symbol instance.

mode (default **normal**) determines how the repeater handles data changes that happen during and after the rendering of the repeater.

- **normal**—A renderer is created for each input row. The repeater's output table is updated at runtime. Each row can be updated multiple times.
- **threaded**—A limited number of renderers are created, based on the **thread** property value. The symbol's **onDone** trigger property causes a renderer to stop processing. The **onDone** property must be added manually and will work automatically if it is a trigger property named onDone. When a renderer stops, the symbol is used on a new input row, so the previous output row is not updated further. The repeater's output table is updated only after all input rows are done. If the input table changes before the output is finished, the new input will be in a pending state. The new processing will start after the first rendering is complete. If the input changes multiple times before it can be processed, only the last input will be in a pending state, and others are skipped.

- **stream**—As with **threaded** mode, a limited number of renderers are created, and the **onDone** property, which must be added manually, stops a renderer. Also similarly to **threaded**, when a renderer stops, its previous rows can't be updated. In contrast to **threaded**, as soon as any row is finished, it is put in the output table. This means that if multiple renderers finish in the same frame, the output table will have more than one row for some input row. The output order is not always the same as the input order. If the input order is changed, the rows are put into a queue. If the input is changed multiple times, all data is merged into the same queue. The queue has a limited size defined by the **streamQueue** property. If **streamQueue** is 0, the queue has unlimited size. Use **stream** mode with Load Value blocks where **qos** is 1 or 3.

thread determines the number of renderers created for a repeater in threaded mode or stream mode. The default is 1.

streamQueue determines the maximum size of the queue for a repeater in stream mode. When the queue is longer, the oldest rows are dropped. The default is 0 (no limit).

keyColumn specifies a column of the input table that identifies each row, in normal mode. When it is set, a renderer is reused for the row with the same key when the input table changes, even if the row moves. Rows whose key is null or repeats an earlier key are skipped.

Renderer Properties

When **symbol** and **data** are defined, renderer properties appear that match the parameters of the symbol.

You can bind columns from the input table to the renderer properties by dragging and dropping the table column headings.

Output Property

The following property of the Repeater block can give output but cannot take input.

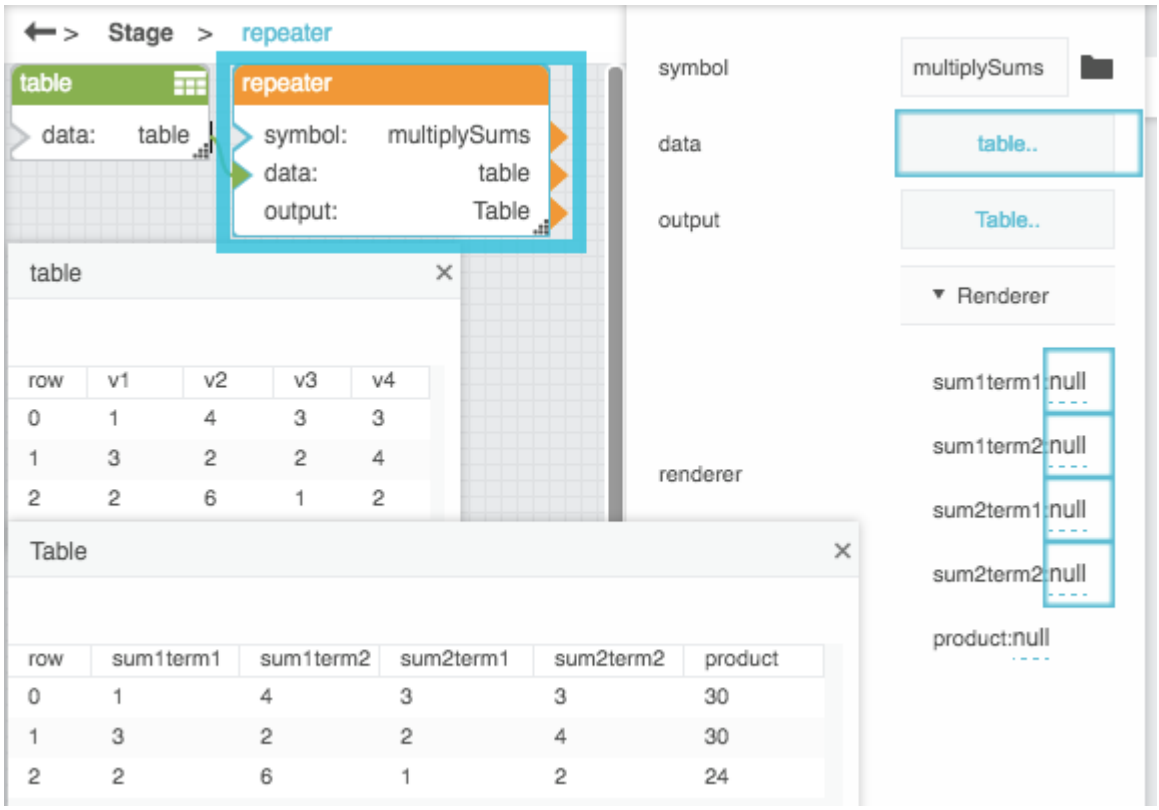
- output *(table)*
- currentQueue *(number)*

output returns a table that lists the symbol parameter values for each instance of the symbol.

currentQueue returns the number of rows waiting in the queue in stream mode. It is not shown on the block.

Example

The following image shows a Repeater block. In this example, the input table's columns are bound to the sum term properties, and the output table includes a product column.



Previous: Drag Size

Next: Catch Error

From:
http://wiki.dglogik.com/ - **DGLogik**

Permanent link:
http://wiki.dglogik.com/dglux5_wiki:dataflow:dataflow_blocks_reference:logic:repeater

Last update: **2026/10/22 13:26**