

# Package: blockr.dag (via r-universe)

June 4, 2026

**Title** A Directed Acyclic Graph Extension for 'blockr'

**Version** 0.1.2

**Description** Building on the docking layout manager provided by 'blockr.dock', this provides an extension that allows for visualizing and manipulating a 'blockr' board using a DAG-based user interface powered by the 'g6R' graph visualisation HTML widget.

**License** GPL (>= 3)

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**Imports** blockr.core (>= 0.1.2.9000), blockr.dock (>= 0.1.1), shiny, g6R (>= 0.6.0), jsonlite, htmltools

**Remotes** BristolMyersSquibb/blockr.core

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0), roxy.shinylive, webshot2, quarto, cyclocomp, shinytest2, colorspace

**Config/testthat/edition** 3

**URL** <https://bristolmyerssquibb.github.io/blockr.dag/>

**VignetteBuilder** quarto

**Config/pak/sysreqs** cmake make libuv1-dev zlib1g-dev

**Repository** <https://bristolmyerssquibb.r-universe.dev>

**Date/Publication** 2026-06-04 15:17:55 UTC

**RemoteUrl** <https://github.com/bristolmyerssquibb/blockr.dag>

**RemoteRef** HEAD

**RemoteSha** 4c886e108018aa94b0f0e8c515b5aae703227588

## Contents

new_context_menu_entry . . . . .	2
new_dag_extension . . . . .	3
new_graph . . . . .	4
new_toolbar_item . . . . .	5

<b>Index</b>	<b>7</b>
--------------	----------

---

new\_context\_menu\_entry  
*Context menu functions*

---

### Description

Functions for creating and working with context menu entries.

### Usage

```
new_context_menu_entry(
  name,
  js,
  action = NULL,
  condition = TRUE,
  id = tolower(gsub(" +", "_", name))
)
```

```
is_context_menu_entry(x)
```

```
context_menu_items(x)
```

### Arguments

name	Name of the context menu entry.
js	JavaScript code to execute when the entry is selected.
action	Action to perform when the entry is selected.
condition	Condition to determine if the entry should be shown.
id	Unique identifier for the context menu entry. Inferred from name if not provided
x	Object

### Details

new\_context\_menu\_entry() Creates a new context menu entry with the specified name, JavaScript code, action function, and display condition.

is\_context\_menu\_entry() Tests whether an object is a valid context menu entry.

`context_menu_items()` Generic function to extract context menu items from various objects like dock extensions, boards, or lists.

The `context_menu_items.dag_extension()` method provides the following actions:

- Create link - Creates connections between workflow nodes.
- Remove block - Removes individual blocks from the workflow.
- Remove link - Removes connections between workflow nodes.
- Append block - Adds a new block after the selected node.
- Create stack - Creates a new workflow stack.
- Remove stack - Removes an entire workflow stack.
- Edit stack - Opens stack editing interface.
- Add block - Adds a new block to the canvas.

### Value

`new_context_menu_entry()` A context menu entry object of class "context\_menu\_entry" containing condition, action, and js functions, with name and id attributes.

`is_context_menu_entry()` TRUE if x is a context menu entry, FALSE otherwise.

`context_menu_items()` A list of context menu items for the given object.

---

<code>new_dag_extension</code>	<i>DAG extension</i>
--------------------------------	----------------------

---

### Description

Visualizes the DAG (directed acyclic graph) underlying a board and provides UI elements to manipulate the board.

### Usage

```
new_dag_extension(graph = NULL, ...)
```

### Arguments

<code>graph</code>	A graph object (or NULL).
<code>...</code>	Forwarded to <code>blockr.dock::new_dock_extension()</code> .

### Value

A `dag_extension` object that extends the dock extension system for visualizing and manipulating DAG workflows.

---

 new\_graph

*Graph object*


---

### Description

R list-based representation of a g6 graph object.

### Usage

```
new_graph(nodes = list(), edges = list(), combos = list())
```

```
is_graph(x)
```

```
graph_nodes(x)
```

```
graph_edges(x)
```

```
graph_combos(x)
```

```
as_graph(x, ...)
```

```
## S3 method for class 'graph'
```

```
as_graph(x, ...)
```

```
## S3 method for class 'list'
```

```
as_graph(x, ...)
```

### Arguments

nodes	Graph nodes (i.e. board blocks).
edges	Graph edges (i.e. board links).
combos	Node groups (i.e. board stacks).
x	Object to test or convert.
...	Generic consistency.

### Details

`new_graph()` Creates a new graph object with the specified nodes, edges, and combos.

`is_graph()` Tests whether an object is a valid graph object.

`graph_nodes()` Extracts the nodes component from a graph object.

`graph_edges()` Extracts the edges component from a graph object.

`graph_combos()` Extracts the combos component from a graph object.

`as_graph()` Generic function to convert objects to graph format.

**Value**

new\_graph() A graph object of class "graph" containing nodes, edges, and combos lists.  
 is\_graph() TRUE if x is a graph object, FALSE otherwise.  
 graph\_nodes() A list of graph nodes.  
 graph\_edges() A list of graph edges.  
 graph\_combos() A list of graph combos (node groups).  
 as\_graph() A graph object converted from the input.

---

new\_toolbar\_item      *Toolbar item functions*

---

**Description**

Functions for creating and working with toolbar items for the DAG interface.

**Usage**

```

new_toolbar_item(id, icon, js, action = NULL, tooltip = NULL)

is_toolbar_item(x)

toolbar_items(x)

```

**Arguments**

id	Unique identifier for the toolbar item.
icon	Name of an icon to show in the toolbar.
js	JavaScript code to execute when the entry is selected.
action	Action to perform when the entry is selected.
tooltip	Optional tooltip text for the entry.
x	Object

**Details**

new\_toolbar\_item() Creates a new toolbar item with the specified id, icon, JavaScript code, action function, and tooltip text.

is\_toolbar\_item() Tests whether an object is a valid toolbar item.

toolbar\_items() Generic function to extract toolbar items from various objects like dock extensions, boards, or lists.

The toolbar\_items.dag\_extension() method provides the following actions:

- Zoom in - Increases the graph zoom level.
- Zoom out - Decreases the graph zoom level.

- Auto fit - Automatically fits the entire graph within the viewport.
- Layout - Reapplies the graph layout algorithm to reorganize nodes.
- Add block - Opens interface to add a new block to the workflow.
- Add stack - Creates a new workflow stack.
- Remove selected - Removes currently selected elements from the graph.

**Value**

`new_toolbar_item()` A toolbar item object of class "toolbar\_item" containing action and js functions, with id, icon, and tooltip attributes.

`is_toolbar_item()` TRUE if x is a toolbar item, FALSE otherwise.

`toolbar_items()` A list of toolbar items for the given object.

# Index

`as_graph (new_graph)`, 4

`blockr.dock::new_dock_extension()`, 3

`context_menu_items`  
    `(new_context_menu_entry)`, 2

`graph_combos (new_graph)`, 4

`graph_edges (new_graph)`, 4

`graph_nodes (new_graph)`, 4

`is_context_menu_entry`  
    `(new_context_menu_entry)`, 2

`is_graph (new_graph)`, 4

`is_toolbar_item (new_toolbar_item)`, 5

`new_context_menu_entry`, 2

`new_dag_extension`, 3

`new_graph`, 4

`new_toolbar_item`, 5

`toolbar_items (new_toolbar_item)`, 5