Mouse Event

Block Group:	Logic
Icon:	◎

Listens for the specified mouse event, and returns the location of the cursor when the event occurs.

As an advanced use, this block can also be used to record information when dragging text, files, or elements onto an element.

For information on using dataflow blocks, see Dataflow.

Input/Output Properties

These properties can take input and give output.

- input *(event)*
- active *(boolean)*
- fastResponse *(boolean)*

input specifies the event that triggers this block. Make sure that you have selected the correct parent element for the event, and then find the event in the Advanced properties and bind it to this property. Examples include:

- **onClick** – The user clicks the element.
- **onClickOff** – The user clicks the element (paired with onClickOn).
- **onClickOn** – The user clicks the element (paired with onClickOff).
- **onDoubleClick** – The user double-clicks the element.
- **onMouseDown** – The user presses the mouse button over the element.
- **onMouseEnter** – The user moves the cursor onto the element.
- **onMouseLeave** – The user moves the cursor off of the element.
- **onMouseMove** – The user moves the cursor while it is over the element.
- **onMouseMoveGlobal** – The user moves the cursor anywhere on the screen.
- **onMouseUp** – The user releases the mouse button over the element.
- **onMouseUpGlobal** – The user releases the mouse button anywhere on the screen.
- **onMouseWheel** – The user rotates the mouse wheel either up or down.
- **onRightClick** – The user right-clicks on the element.

active sets or returns whether this block is currently listening for the specified event.

fastResponse determines whether the fastResponse feature is enabled. Use this feature when you troubleshoot performance speed issues relating to this block.

- **TRUE**: The execution of this block’s event is slightly prioritized, and the queue is kept clear.

- **FALSE**: This is the default behavior.

Output Properties

These properties can give output. They cannot take input.

- event *(event)*
- pageX *(number)*
- pageY *(number)*
- dragData *(binary)*
- dragSource *(number)*
- fileName *(number)*

event returns the event type that this listener heard.

x returns the distance, in pixels, from the left container boundary of the element to the cursor at the time of the mouse event. Positive values are to the right of the boundary.

y returns the distance, in pixels, from the top container boundary of the element to the cursor at the time of the mouse event. Positive values are below the top container boundary.

pageX returns the distance, in pixels, from the left page boundary to the cursor at the time of the mouse event. Positive values are to the right of the left page boundary.

pageY returns the distance, in pixels, from the top page boundary to the cursor at the time of the mouse event. Positive values are below the top page boundary.

dragData returns the binary data of the file that is dropped onto this element. To use this property, you must do the following:

- Enable the **droppable** Advanced property for this element.
- Set the **dropGroup** Advanced property for this element to `file_png`. You can also replace png with another file extension.

dragSource returns the ID of the element that is dropped onto this element. To use this property, you must do the following:

- Enable the **droppable** Advanced property for this element.
- Enable the **draggable** Advanced property for the element to be dragged.
- Make sure that the **dragGroup** Advanced property is the same for both elements.
- Make sure that the **dragGroup** Advanced property is the same for both elements.

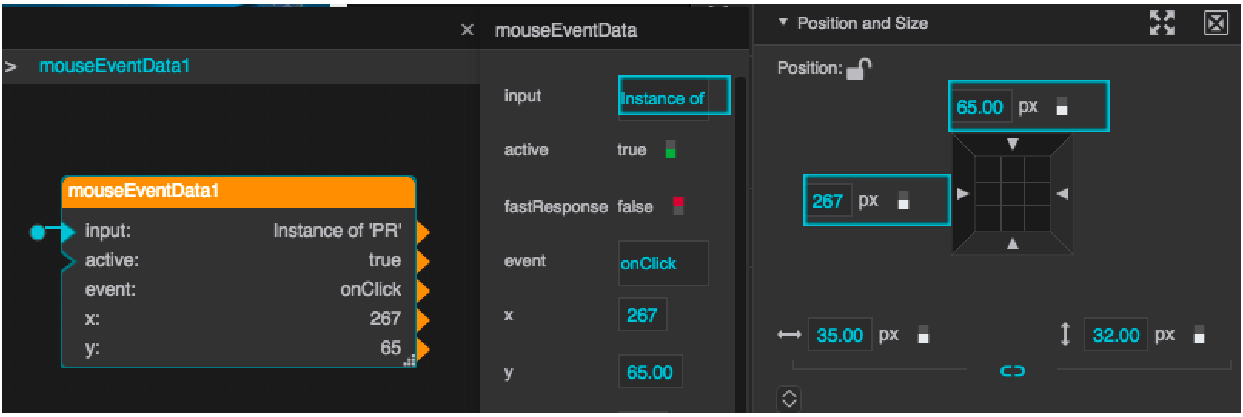
fileName returns the name of the file that is dropped onto this element. To use this property, you must do the following:

- Enable the **droppable** Advanced property for this element.

- Set the **dropGroup** Advanced property for this element to `file_png`. You can also replace png with another file extension.

Model

This is a basic use of the block.

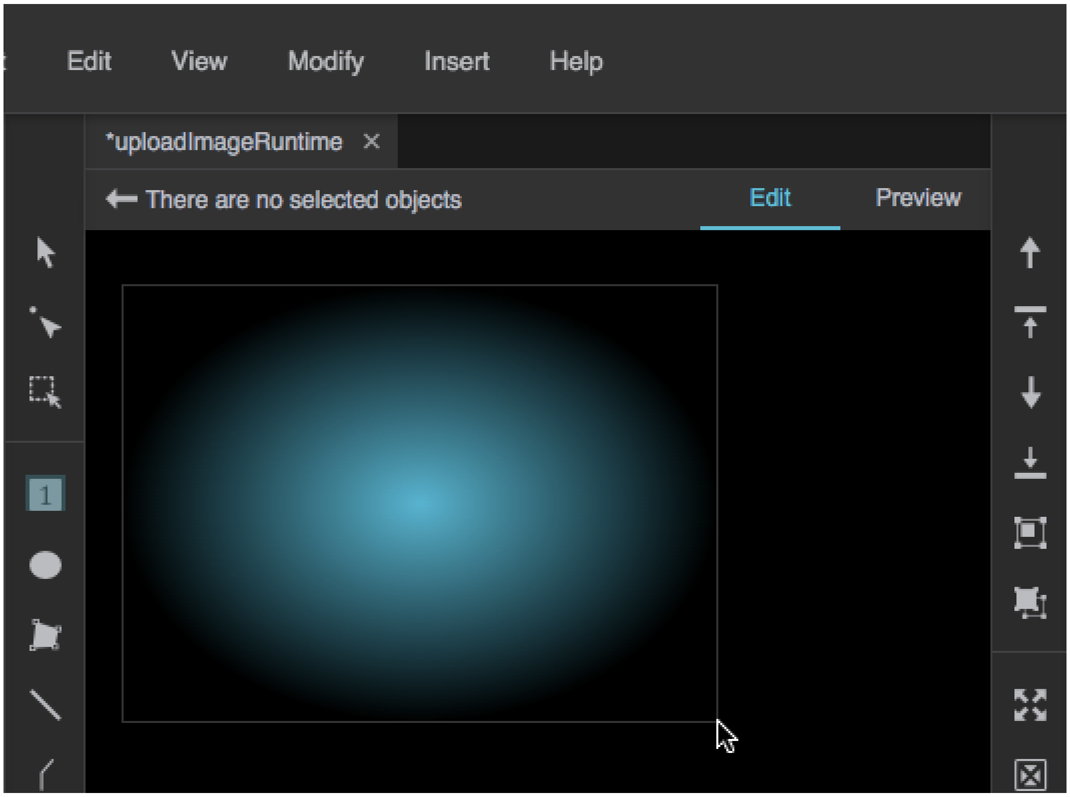
Dataflow Model	Description
 mouseEventData1 input: Instance of 'PR' active: true event: onClick x: 267 y: 65 mouseEventData input Instance of active true fastResponse false event onClick x 267 y 65.00 Position and Size Position: 65.00 px, 267 px 35.00 px, 32.00 px	The **x** and **y** properties of a group's Mouse Event block are bound to the **x** and **y** properties of a child shape. When the mouse enters the group, the shape moves to the mouse location.

Use Case: Uploading Files at Runtime

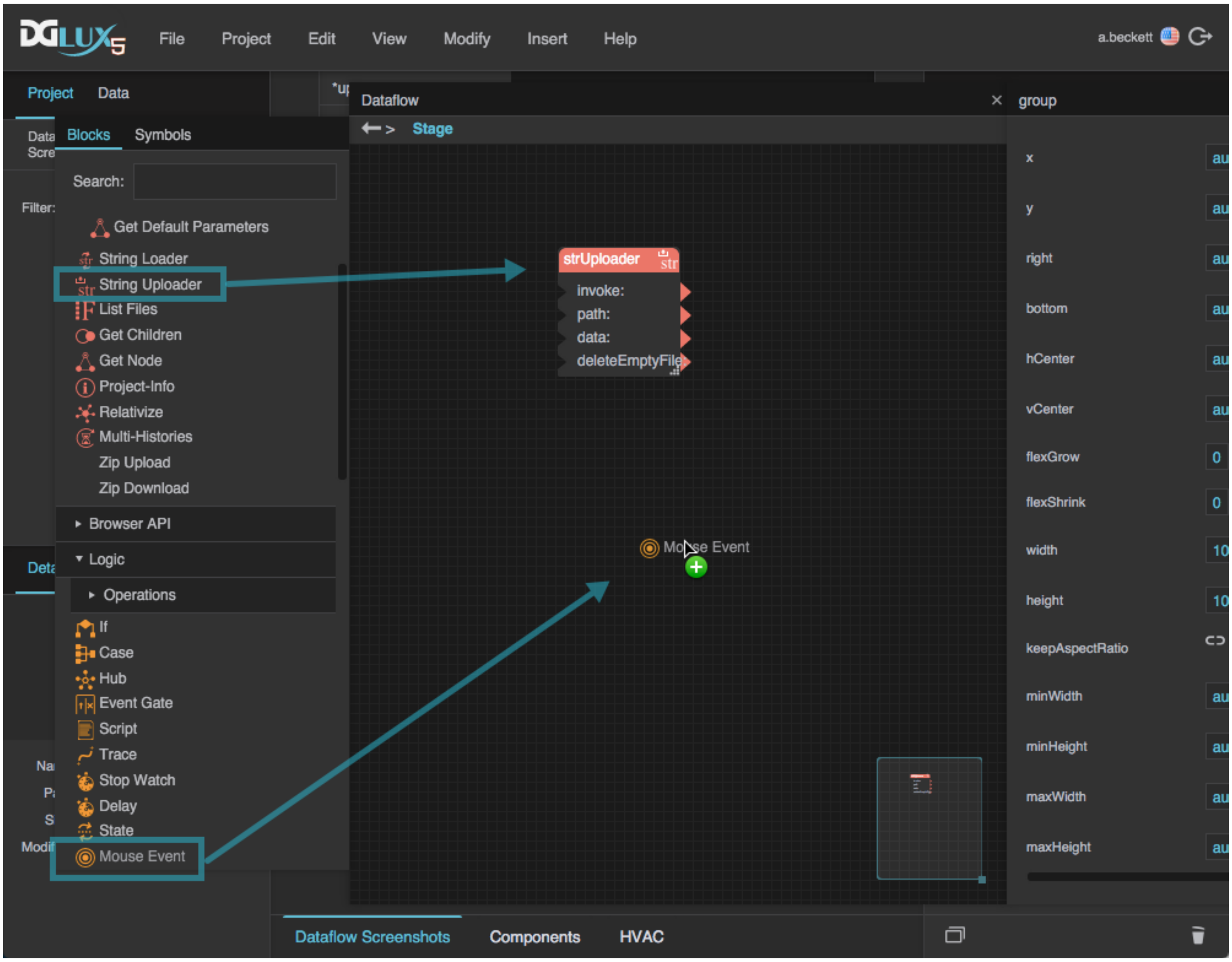
This example shows you how to enable uploading files to the project at runtime.

For steps to create a file input component that allows file upload at runtime, see File Input.

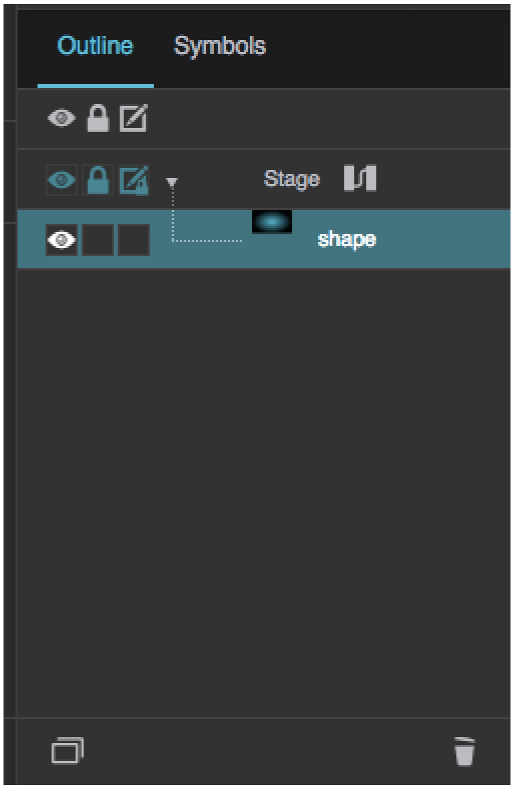
1. Insert a shape.



2. In the Outline, right-click the Stage and select **Dataflow**. Then, add a String Uploader block and a Mouse Event block to the Dataflow window.

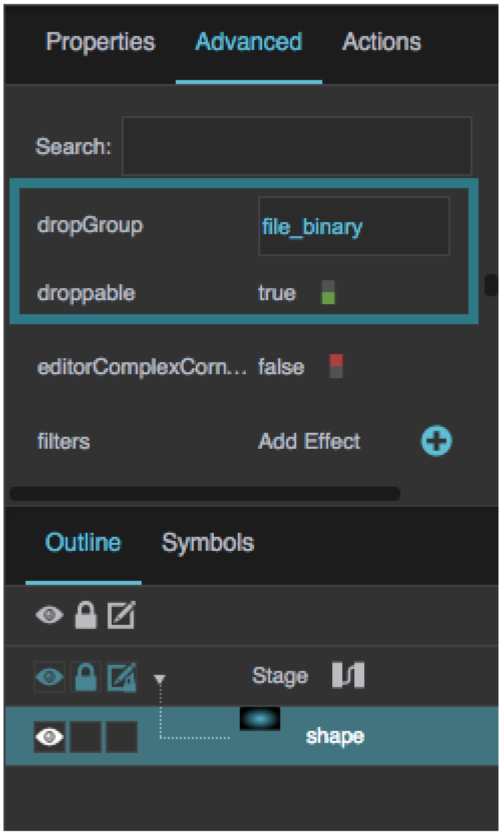


3. In the Outline, select the shape.



4. In the Advanced properties panel:
    1. Set **droppable** to TRUE.

    2. For **dropGroup**, enter `file_binary`.



Search:

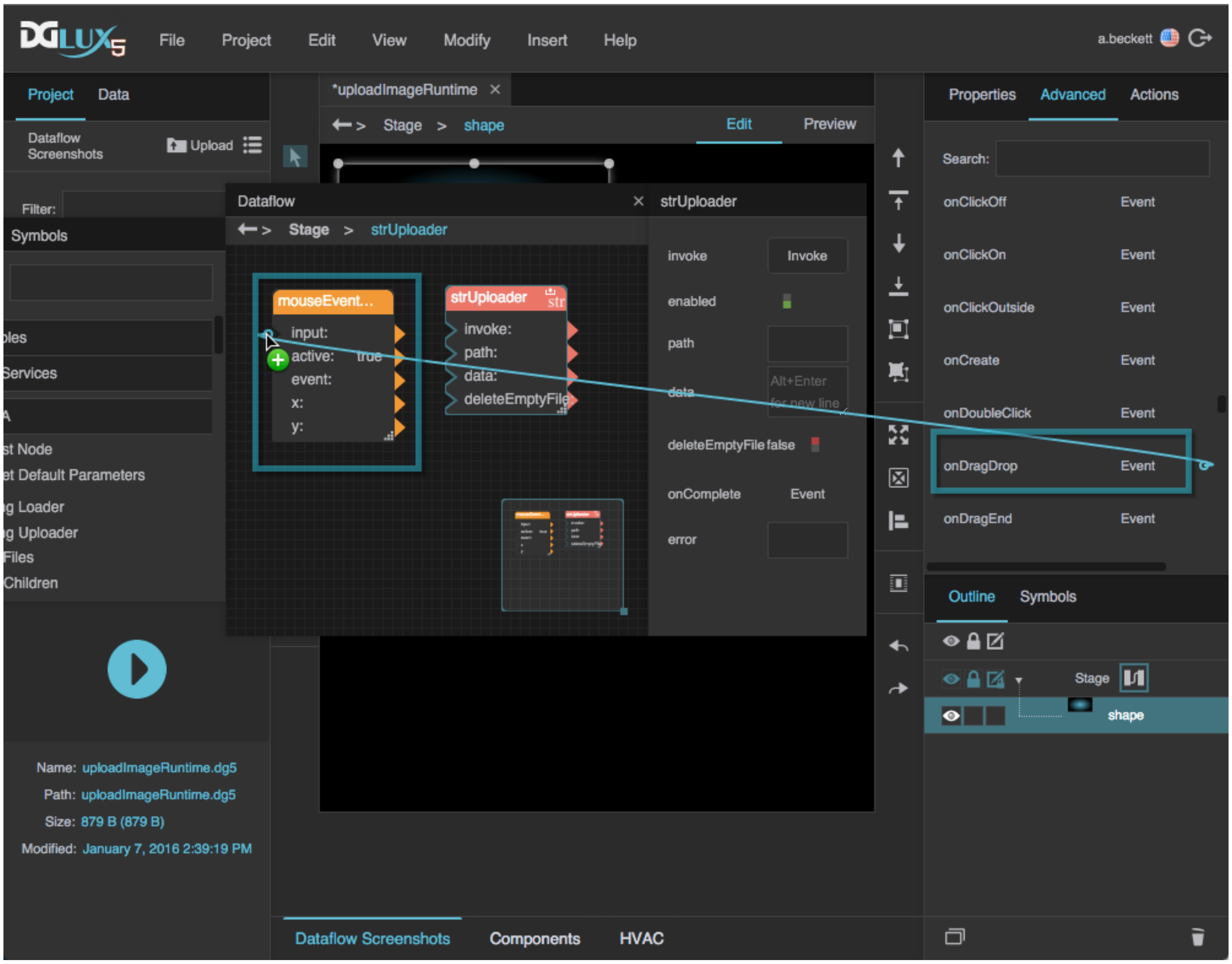
dropGroup file_binary

droppable true

editorComplexCorn… false

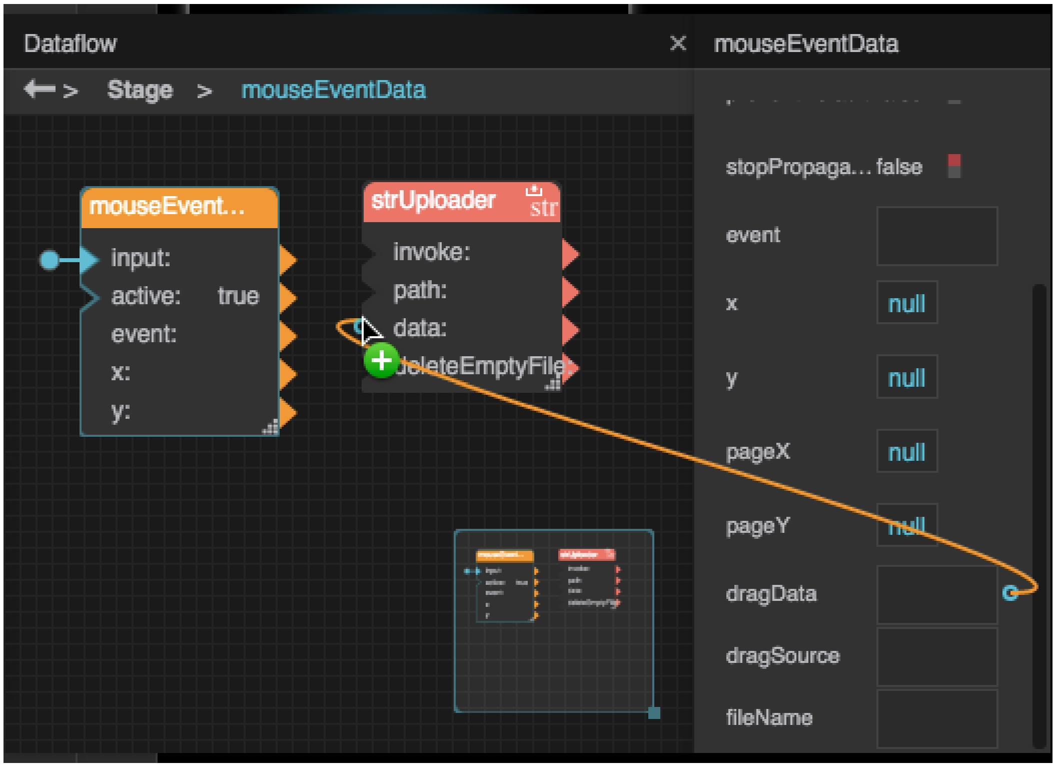
filters Add Effect

3. Find the **onDragDrop** event, and bind it to the **input** property of the Mouse Event block.



Name: uploadImageRuntime.dg5
Path: uploadImageRuntime.dg5
Size: 879 B (879 B)
Modified: January 7, 2016 2:39:19 PM

5. Bind the **dragData** property of the Mouse Event block to the **data** property of the String Uploader block.



stopPropaga... false

event

x null

y null

pageX null

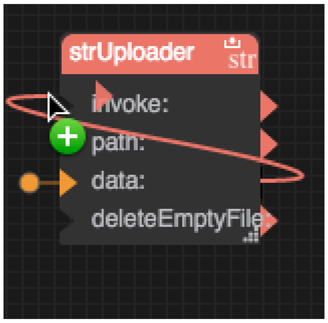
pageY null

dragData

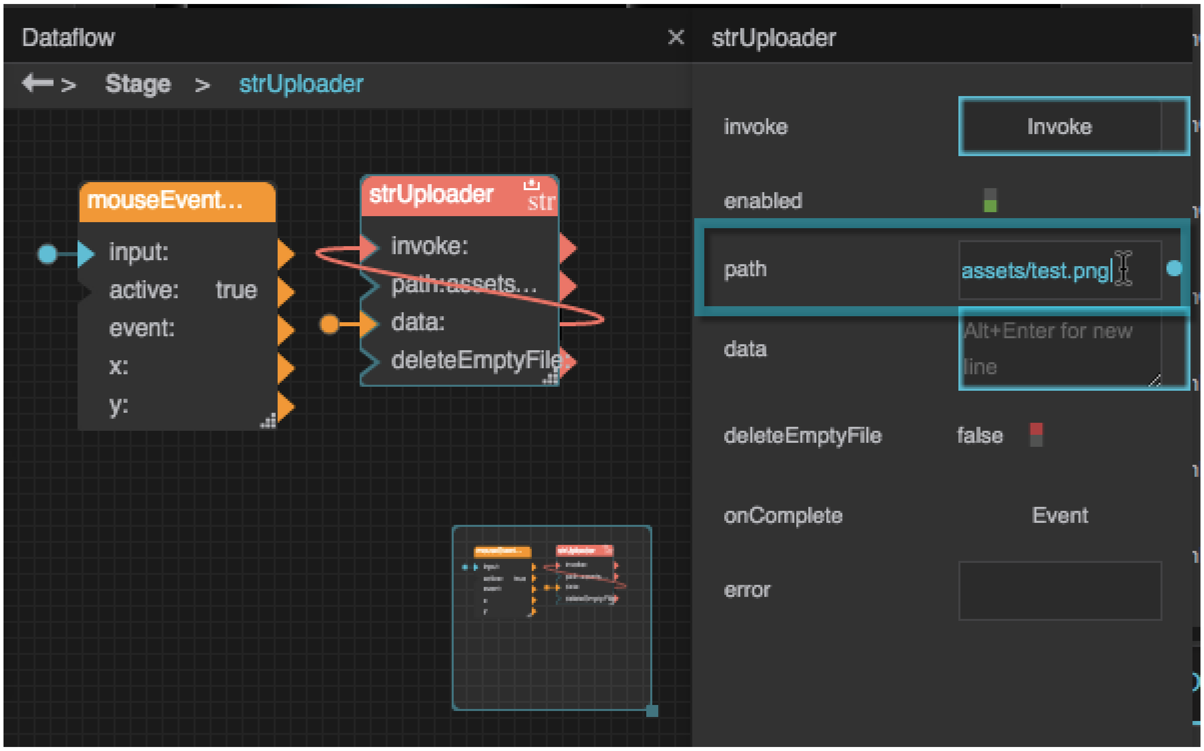
dragSource

fileName

6. Bind the **data** property of the String Uploader block to the **invoke** property of the String Uploader block.



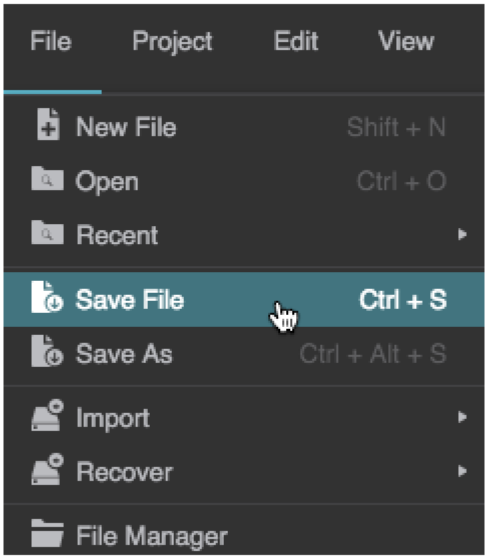
7. For the **path** property of the String Uploader block, add the path of the folder where you want the uploaded file to be stored, including the file name and extension for the uploaded file. For example, enter `assets/test.png`.

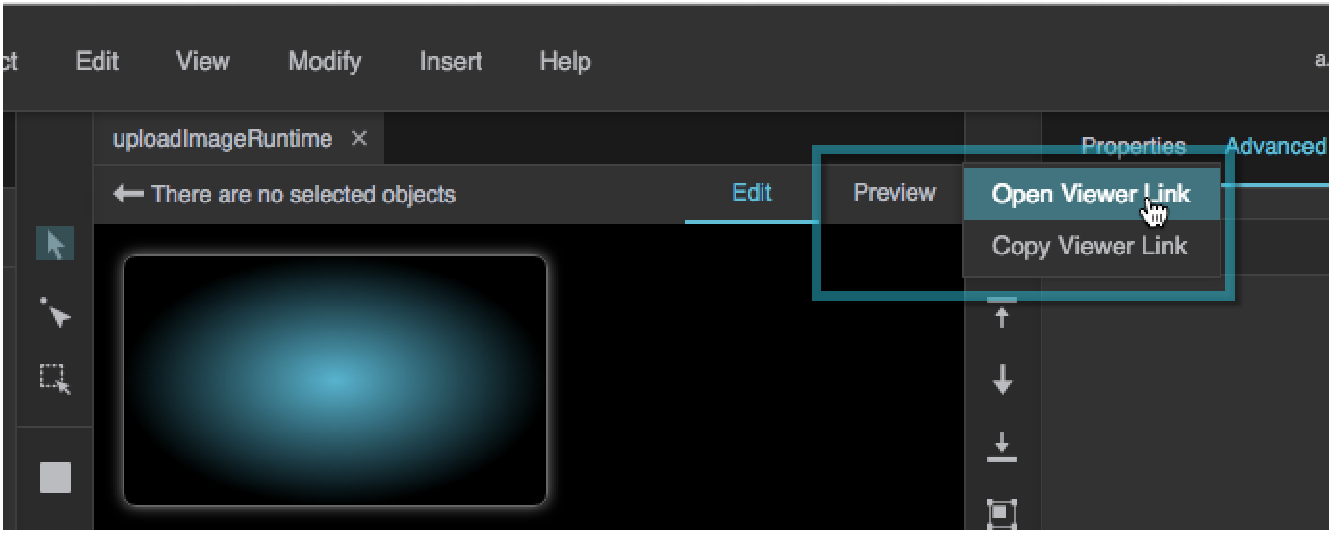


strUploader

invoke	Invoke
enabled	
path	assets/test.png
data	
deleteEmptyFile	false
onComplete	Event
error	

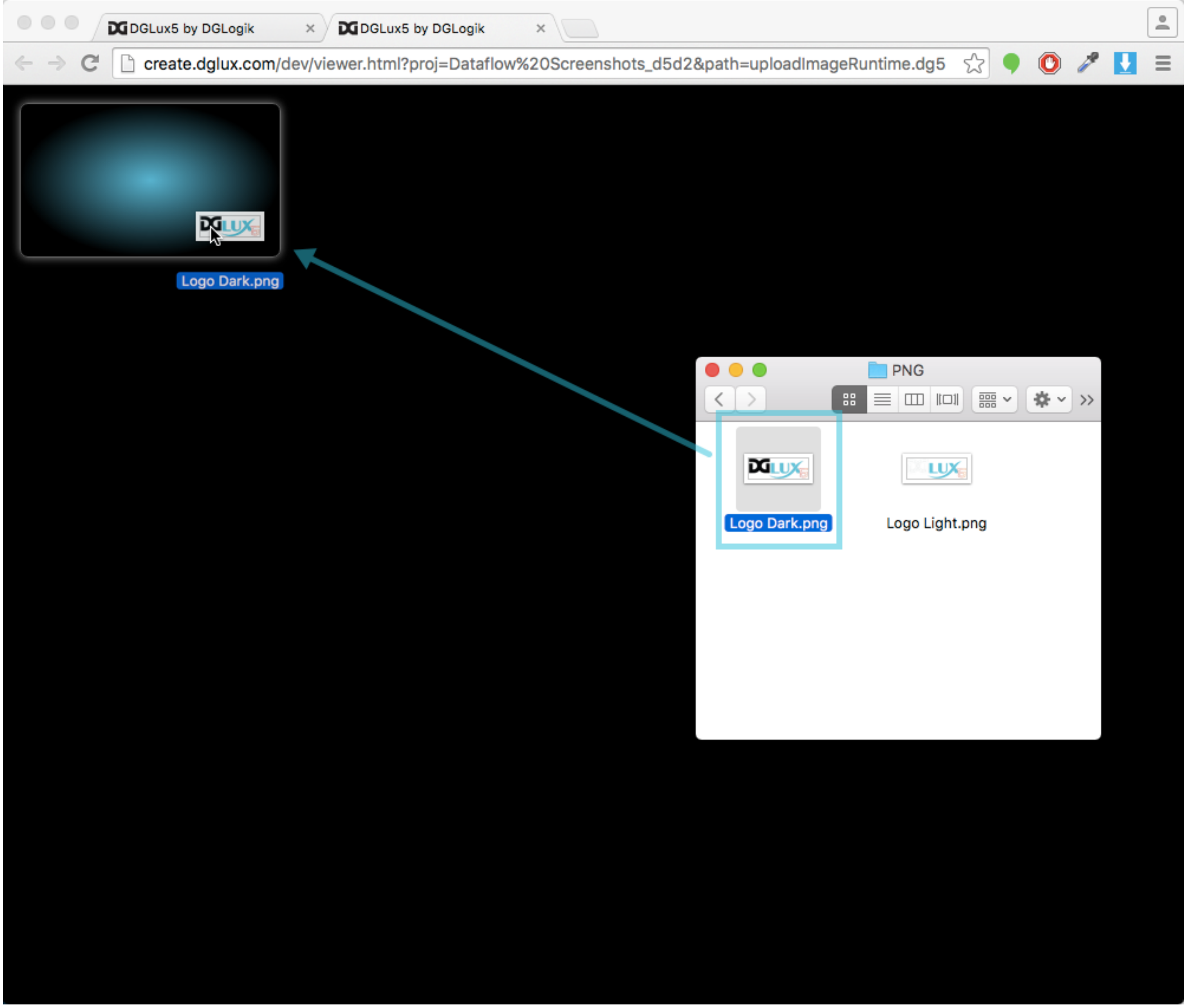
8. Save the page, and then open the Viewer.



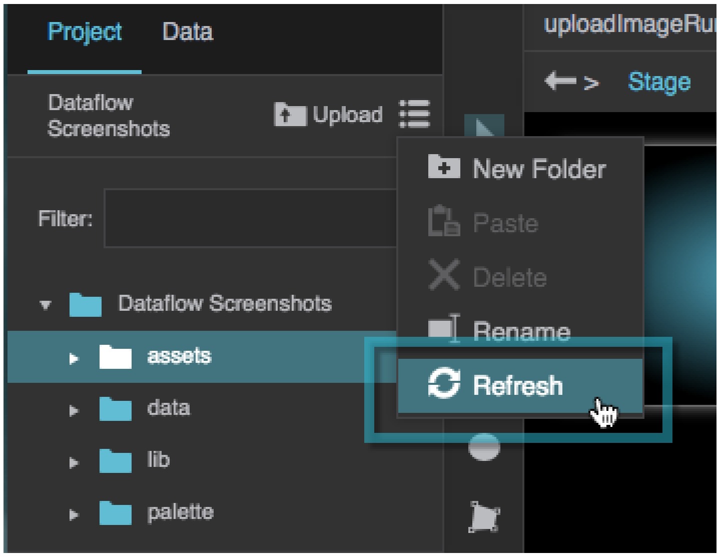


The Viewer opens in a new browser tab.

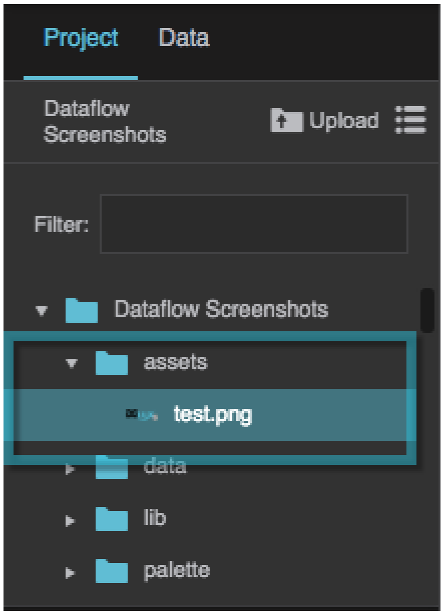
9. To test, drag a PNG file from your computer's file system to the shape.


DGLux5 by DGLogik
DGLux5 by DGLogik
create.dglux.com/dev/viewer.html?proj=Dataflow%20Screenshots_d5d2&path=uploadImageRuntime.dg5
Logo Dark.png
PNG
Logo Dark.png
Logo Light.png

10. In the browser tab that holds the DGLux5 editing editing environment, in the Project panel, select ☰ **Menu** > **Refresh**.



The image that you uploaded is in the specified folder.



More Resources

This thread in the DGLogik Community Forum shows another use case for the block:

- Getting cursor position

Previous: State

Next: Keyboard Event

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

Permanent link:
https://wiki.dglogik.com/dglux5_wiki:dataflow:dataflow_blocks_reference:logic:mouse_event

Last update: **2021/09/20 15:03**