

# String Uploader

Block Group:	Data Services
Block Icon:	

The String Uploader block saves a string as a new file in this project. If a file with the specified name and extension already exists, this block replaces the contents of the existing file.

For information on using dataflow blocks, see [Dataflow](#).



## Note

You might need to collapse the folder and re-expand it, or save your work and refresh the browser, to see the new file.

## Input/Output Properties

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

- *invoke* (*trigger*)
- *enabled* (*boolean*)
- *path* (*string*)
- *data* (*string*)
- *deleteEmptyFile* (*boolean*)

**invoke** causes the string to be uploaded. The **invoke** property works only if **enabled** is set to TRUE.

**enabled** specifies whether the String Uploader block is enabled.

- **TRUE**—The string is uploaded when this block is invoked.
- **FALSE**—Nothing happens when this block is invoked.

**path** specifies where to upload the string, relative to the root of this project. Must include a file name and extension.

- To save to the root, enter `/<filename>.<extension>`.
- To save to the assets folder, enter `assets/<filename>.<extension>`.

**data** holds the string that gets uploaded.

**deleteEmptyFile** specifies what happens when the **data** property is null.

- **TRUE**: If **data** is null, no new file is created. If the file already exists, it is deleted.
  - **FALSE**: If **data** is null, a file is created when this block is invoked. The file is empty. If the file already exists, its contents are deleted but the file remains.
- 

## Output Properties

The following properties of the String Uploader block can give output but cannot take input.

- `onComplete` (*event*)
- `error` (*string*)

**onComplete** fires when the string has been uploaded.

**error** returns the error message, if any.

---

## Example

The following image demonstrates a typical use of the String Uploader block. In this example, when the String Uploader block is invoked, a file named `new_file.txt` is created.

The screenshot shows a Dataflow application interface. On the left, there is a 'Project Data' sidebar with a file explorer showing folders 'data', 'lib', and 'new\_folder', and a file 'new\_file.txt'. Below the file explorer is a 'Details' section for 'new\_file.txt' showing its contents: 'Item 1', 'Item 2', and 'Item 3'. In the center, a 'Blocks' menu is open, listing various categories like Variables, Data Services, Browser API, Logic, String Operations, Math Operations, Number Formatting, Statistical Functions, Trigonometric Functions, Table Operations, and Date Time Operations. On the right, a 'String Uploader' block is highlighted in a red box. The block's configuration is as follows:

```
strUploader  
invoke: Instance of 'IB'  
path: /new_folder/new_file.txt  
data: Contents of the new file: • Item 1 • Item 2 • Item 3  
deleteEmptyFile:
```

## More Resources

The use case on the following page demonstrates how to create a file input component that allows the user to upload files at runtime.

- [File Input](#)

[Previous: String Loader](#)

[Next: List Files](#)

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

Permanent link: [https://wiki.dglogik.com/dglux5\\_wiki:dataflow:dataflow\\_blocks\\_reference:data\\_services:string\\_uploader](https://wiki.dglogik.com/dglux5_wiki:dataflow:dataflow_blocks_reference:data_services:string_uploader)

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

