

Sort files in job



Sort files in job is a default Switch app that provides a mechanism for "sorting" the files inside a job *folder*, in other words to perform the following two steps:

1. Determine the desired sorting order of the files based on their current filename.
2. Rename the files so that they alphabetically collate in the desired sorting order.

This is useful to prepare the job for tools that process files in alphabetical order, for example merging pages into a single document.

Sort files in job ignores any nested subfolders and their contents (that is, it works only on the files immediately inside the job folder). Individual job files are passed along unchanged.

Keywords

Keywords can be used with the search function above the Elements pane.

The keywords for the **Sort files in job** element are:

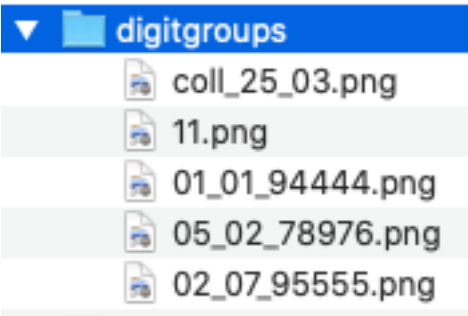
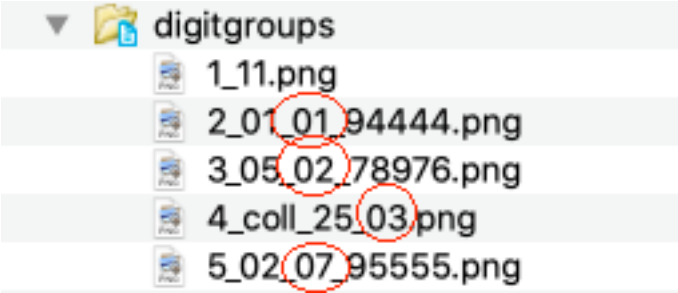

- remove
- add
- replace
- order
- filename
- name
- extension


Connections

Sort files in job allows only a single outgoing connection.

Properties

| Property | Description |
|--------------|---|
| Element type | The flow element type: Sort files in job. This property is useful to identify renamed flow elements. It cannot be changed. |
| Name | The name of the flow element displayed in the canvas |

| Property | Description |
|--------------|--|
| Description | A description of the flow element displayed in the canvas. This description is also shown in the tooltip that appears when moving your cursor over the flow element |
| Sort key 1 | <p>Determines the first sort key:</p> <ul style="list-style-type: none"> • None • Number: the Nth distinct group of consecutive digits • Segment: the segment specified by start index and length (to sort by a part of the filename) • Search: the segment matching a regular expression • Extension: the filename extension |
| Number index | <p>The one-based index of the intended digit group; automatic means "use the smallest number" of all digit groups</p> <p>For example, if number index = 2, the files will be sorted by the second digit group.</p> <p>Example input folder:</p>  <p>Example output:</p>  <p> Note: If a filename does not have a digit group that matches the number index (e.g. if the number index</p> |

| Property | Description |
|-----------------|--|
| | is 2 and the filename contains only 1 digit group), the filename concerned is put at the top of the list (like '11.png' in this example). |
| Start index | The one-based start index of the intended filename segment. For example, to sort files by the three first characters of the filename, set the start index to 1 and the length to 3. |
| Length | The length of the intended filename segment; zero means until the end of the filename, excluding extension. For example, to sort files by the three last characters of an 8 character filename, set the start index to 5 and the length to 8. |
| Search pattern | A regular expression that matches the intended segment; if there is no match the sort key is empty |
| Sort as numbers | <p>If set to no, the filename segments specified in the previous properties are sorted as regular text strings</p> <p>If set to yes, the segments are first padded with leading zeros (to the maximum length in the set) so that they collate as one would expect for numbers (note that this procedure works for hexadecimal numbers in addition to regular decimal numbers)</p> <div>  Note: This property setting is only relevant in case some of the filename segments have different lengths. </div> <p>Example:</p> <ul style="list-style-type: none"> Input folder contains filenames like A5, A06, C2, C12, C123, ... Sort as numbers set to No: The digits are treated as strings. The letters will be put in alphabetical order and the digits will be sorted digit by digit, resulting in the following sort order: A06, A5, C12, C123, C2. Sort as numbers set to Yes: In the background leading zeros are added to make sure the digit groups are sorted correctly. This results in the following sort order: A5, A06, C2, C12, C123. |
| Sort key 2 | <p>The second and third sort keys; these have the same subordinate properties as the first sort key</p> <p>If two or more filenames have identical sort key values, they are sorted alphabetically</p> |
| Sort key 3 | |
| Rename scheme | <p>Determines the rename action taken to make files collate:</p> <ul style="list-style-type: none"> Add prefix Add prefix and suppress digits: each digit group in the original filename is replaced by an underscore |

| Property | Description |
|----------------|--|
| | <ul style="list-style-type: none">• Replace filename (leave extension)• Replace filename and extension |
| Prefix pattern | A pattern for the prefix; "count" is replaced by a number with an appropriate number of digits (depending on the number files) |