StringWizard

Table of Contents

1. Introduction

StringWizard offers string manipulation to change case, replace unwanted characters (such as space, diacritics, ligatures, trademarks or reserved characters), and truncate strings.

This is particularly useful when there is a need of string standardization, such as making filename suitable for the web or making it compatible between platforms (for file transfer or archive).

1.1. Versions

The following is a short version overview:

- **Version 1**: initial version of the app.

1.2. Compatibility

Enfocus Switch 2023 Fall Edition and higher for Windows and Mac.

StringWizard will work on any Switch configuration, but the Metadata module is required to used private data as input and/or output.

2. How to use

StringWizard needs an incoming connection and a single outgoing connection.

StringWizard can perform several string conversions, which can be combined so that multiple changes can be processed simultaneously. You can build your own configuration by enabling or disabling any property of the app.

Modifications are applied in the following order (from top to bottom of the app properties):
Convert diacritics and ligatures > Trademarks > Case > Space > Reserved characters > Truncate

2.1. String conversion

StringWizard uses different types of input and output to solve many use cases of string conversion, e.g. using a private data input to rename the incoming job, process the incoming jobname and save the result as a private data, or convert the content of a private data.

2.1.1. Input type

There are 3 types of input that can be sent to StringWizard:

- **Incoming jobname**
- **Private data**
- **Text**

By default, this property is set to 'Incoming jobname'.

2.1.1.1. Incoming jobname

If the input string is the incoming jobname, another property is requested:

- **Jobname proper**
  Only the jobname proper is processed, the extension is kept as is
- **Complete jobname**
  The complete jobname is processed, including the extension

By default, this property is set to 'Jobname proper'.

2.1.1.2. Private data

The input string can be a private data. This private data can be provided inline, using a single-line text with variables or a script expression.

2 properties are requested when the input is a private data:

- **Key**
  The private data key to be used (mandatory)
- **Add incoming job extension on output**
  The incoming job extension can be added or omitted on output (when the output is a renamed job) – default: 'Yes'

2.1.1.3. Text

The input string can be some text. This text can be provided inline, using a single-line text with variables or a script expression.

2 properties are requested when the input is a private data:

- **Text**
  The text to be used for conversion (mandatory)
- **Add incoming job extension on output**
  The incoming job extension can be added or omitted on output (when the output is a renamed job) – default: 'Yes'

2.1.2. Output type

There are 2 types of output that can be provided by StringWizard:

- **Rename incoming job**
- **Private data**

By default, this property is set to 'Rename incoming job'.

2.1.2.1. Rename incoming job

If the selected output is 'Rename incoming job', the incoming job is going to be renamed with the processed string from the input.

2.1.2.2. Private data

If the selected output is 'Private data', another property is requested:

- **Key**
  The private data key to be used (mandatory) – default: 'aazappCleanedString'

2.1.3. Use case examples

Here is how to set up input and output to serve the following use cases:

- **Rename the incoming job using the job's name**
  Input: 'Incoming jobname' / Output: 'Rename incoming job' – choose between jobname proper and complete jobname
- **Rename the incoming job using a (cleaned) private data**
  Input: 'Private data' / Output: 'Rename incoming job' – don't forget to add incoming extension if you want to keep it on the outgoing job
- **Store the cleaned jobname as a private data**
  Input: 'Incoming jobname' / Output: 'Private data' – the cleaned jobname will be added as a private data to the job, but the job keeps its original name
- **Clean a private data**
  Input: 'Private data' / Output: 'Private data' – use the same key to update the private data, another name to create a new cleaned private data

2.2. Convert diacritics and ligatures

StringWizard can convert diacritics (see 3.1 for definition) and ligatures (see 3.2 for definition).

If set to 'Yes', diacritics will be converted to their equivalent without alteration:

- éléphant ➜ elephant

Ligatures will also be replaced by their equivalent:

- œuf ➜ oeuf

By default, this property is set to 'Yes'.

2.3. Convert trademarks

StringWizard can convert trademarks (see 3.3 for definition).

If set to 'Yes', the trademark will be replaced by a sequence of characters:

- © ➜ (C)
- ® ➜ (R)
- ™ ➜ (TM)

By default, this property is set to 'No'.

2.4. Convert case

The case of the processed string can be changed. The different values for this property are:

- **Keep**
  This is the String to Modify ➜ This is the String to Modify (no change)
- **lowercase**
  This is the String to Modify ➜ this is the string to modify
- **UPPERCASE**
  This is the String to Modify ➜ THIS IS THE STRING TO MODIFY

By default, this property is set to 'lowercase'.

2.5. Convert space

Spaces in the processed string can be kept, removed or replaced. The different values for this property are:

- **Keep**
  This is the String to Modify ➜ This is the String to Modify (no change)
- **Remove**
  This is the String to Modify ➜ ThisistheStringtoModify
- **Dash**
  This is the String to Modify ➜ This-is-the-String-to-Modify
- **Underscore**
  This is the String to Modify ➜ This_is_the_String_to_Modify

By default, this property is set to 'Underscore'.

2.6. Convert reserved characters

Reserved characters (see 3.4 for list of reserved characters) in the processed string can be kept, removed or replaced. The different values for this property are:

- **Keep**
  Don't use this! ➔ Don't use this! (no change)
- **Remove**
  Don't use this! ➔ Dont use this
- **Dash**
  Don't use this! ➔ Don-t use this-
- **Underscore**
  Don't use this! ➔ Don_t use this_

By default, this property is set to 'Underscore'.

2.7. Truncate string

The string to be processed can be truncated. When truncation is set to 'Yes', the 'Keep' property is displayed and must be filled in. Entering a positive number will truncate the string from the beginning, a negative number from the end of the string.

- **7**
  "Keep some characters" ➔ "Keep so"
- **-6**
  "Keep some characters" ➔ "acters"

By default, this property is set to 'No'.

Note that:

- One space counts as one character
- If the number entered is greater than the number of characters of the string, the string will not be truncated!
- Keep in mind that converting ligatures, trademarks (and some diacritics) will increase the number of characters (e.g. œ ➔ oe). This is the reason why the truncation is done at the end of the process to avoid getting a wrong number of requested characters

3. Glossary

3.1. Diacritic

Diacritic is a glyph added to a letter that somehow alters its pronunciation, and which appears above, below, within or between letters — French uses five diacritics: the grave, acute, and circumflex accents, as well as the cedilla and dieresis.

3.2. Ligature

Ligature is a link between two printed letters

3.3. Trademark

Trademark us a symbol, word, or words legally registered or established by use as representing a company or product.

StringWizard can convert the following symbols:

©	®	™

3.4. Reserved character

Reserved character is a character that is not allowed and/or not safe for web or OS filenames.

Currently, the reserved characters captured by StringWizard are:

,	;	:	.	!	?	*	#	@	%	=	+	'	'	"	`	^	~	&	$	<	>	[	]	{	}			\	/

4. More information

4.1. About us

This app was created by **aazapp**. You can find more information about us here: **https://aaz.app**. We're geeks about topics such as print, PDF, prepress, preflight, automation, and a good deal more. We like to share our expertise with others, sometimes by doing paid professional services projects and other times by bundling our knowledge into (often free) Switch apps.

When we build an app and lovingly craft the code, branding, documentation, and everything else that goes into it, it's because we like it and think it will be helpful for other people as well—people like you.

4.1.1. Is it?

Rating our app on the Enfocus app store is quick and easy, but it makes us very happy. It feels like a reward for the time we have invested in the app. So please visit the app store (https://www.enfocus.com/en/appstore/overview) and take a second to rate it.

4.1.2. Found a problem?

We don't know your Switch flows, so we can't guarantee that our apps will play nicely with them (however much time we invest in testing before releasing them). Please don't hesitate to reach out to us and ask questions, give feedback, and make suggestions about the apps you're using or about other things that would be useful for you in Switch.

Just send us an email at **hello@aaz.app**. We'll be with you… ASAP (obviously).

4.2. Sushi

Oh… one more thing!

If you're using our free apps and they are valuable to you, please consider donating a small amount to us. We (the geeks behind **aazapp**) have a tradition of having sushi together in different cities worldwide, and some pocket money to do so is always appreciated!