

# Read any slot value

## Introduction

You can use the export tool to read any slot in the station and even to transform some BFormat syntax into resolved values. The syntax follows the [SFormat](#). SFormat stands for SuperFormat (and is heavily used with Active-Framework). It's super powerful but you can only use the "child" and "slot" methods in this case.

It will give the value in the targeted slot by the SFormat. This column can only be used at the export.

- In the sub column name, you must put a SFormat ⚠ a simple BFormat won't work. (to encapsulate a BFormat just use {%....%})
- The target slot must exist in the station
- If the targeted slot is not a BFormat or a SFormat the value won't be resolved
- If the targeted slot is a BFormat or a SFormat, it will be resolved by default.
- If the unresolveoption is added, a BFormat or a SFormat slot won't be resolved during the export

## Example

Let's see how we can read and resolve the source name of any alarm extension: `info:{child('alarm:AlarmSourceExt').slot('alarmSourceName')}`

The first column exports the resolved value, the second one the raw BFormat

Alarm Source Name	Alarm Source Name
Read the resolved SourceName of any alarm extensions	Read the resolved SourceName of any alarm extensions
<code>info:{child('alarm:AlarmSourceExt').format('sourceName')}</code>	<code>info:{child('alarm:AlarmSourceExt').slot('sourceName')}{unresolve}</code>
Valve command 1	%parent.displayName%
Valve command 2	%parent.displayName%

## Column syntax

This column is a [dynamic column](#): its name (in the first row) doesn't matter. Here we named the columns "Alarm Source Name" but it could be anything else. Put any name you want.

Define the name of the subcolumn with the following format:

- `info:SFormatExpression`

You can add at the end the option `{unresolve}` if you want the BFormat or SFormat original syntax.

## Accepted value

- None (only for export)

## Try it yourself !

Here is the file used in the above export example. Download it, import it a first time to have the components then try exporting it and see the results. The values in the cell will depend on the values in your workbench.

[readAnySlotValue.xlsx](#)

## Advanced feature

You can also add this column inside a Column targeting a [subComponent](#) (History, Alarm, SubComponent or InfoSource column). To do so, simply add a subColumn with the "info:..." syntax to the desired subComponent.

Now, the base will be the SubComponent itself. You can use it for shorter syntaxes.

Example: Here we put the info subColumn in the Alarm column, so we can directly fetch the information. No need to search for the child in the SFormat

Alarm			
Create/Update an alarm extension			
model	name	info:{format('sourceName')}	info:{slot('sourceName')}{unresolve}
		Valve command 1	%parent.displayName%
		Valve command 2	%parent.displayName%

## Try it yourself !

Here is the file used in the above export example. Download it, import it a first time to have the components then try exporting it and see the results. The values in the cell will depend on the values in your workbench.

[readAnySlotValue-adv.xlsx](#)