

StatusStringToFacets

Description

The StatusStringToFacets component allows you to transform a StatusString into Facets.

Implementation

- Drag and drop the component into the station
- Connect the input property

example: Here, the value in the InputSlot slot has been converted into Facets in the Output slot

StatusStringToFacets (Status String To Facets)	
Status	{ok}
Fault Cause	
Output	showDate=true >> ⌚
Input	showDate=b:true {ok} @ def ▼

Properties

- *Status*: Indicates the status of the component. Can be {ok} or {fault}.
- *FaultCause*: Gives the error description if applicable
- *Input*: The StatusString used to retrieve the Facets string value
- *Output*: The created facets

Examples:

For any Boolean facet (showDate, showMilliseconds, multiLine, etc), your string should look like this: `<key>=b:<boolean value>`, for example `showDate=b:false`

For any Integer facet (precision, fieldWidth, etc), your string should look like this: `<key>=i:<integer value>`, for example `fieldWidth=i:10`

For any Float facet (min, max, etc), your string should look like this: `<key>=f:<float value>`, for example `max=f:10.0`

For any String facet (trueText, falseText, etc), your string should look like this: `<key>=s:<escaped string value>`, for example `trueText=s:on`

For the maxOverrideDuration facet, your string should look like this: `maxOverrideDuration=r:<time in milliseconds>`, for example `maxOverrideDuration=r:3600000`

For the range facet,

- if you use a frozen enum, your string should look like this: `range=E:<frozen enum TypeSpect>`, for example `range=E:aapah:AaPhpDataTypesEnum`
- otherwise, your string should look like this: `range=E:{<escaped tag>=<ordinal> pairs separated by commas}`, for example `range=E:{first=0,second=1}`

For the Unit facet, it is rather complicated, your string should look like this:

`units=u:<unitName>;<symbol>;<dimension>;*<scale>;+<offset>;` for example: `units=u:square meter;m²;(m2);;`

some rules:

- `<symbol>;` is only added if symbol is different from unitName. The text is not escaped.
- `*<scale>;` is only added if it is different from 0.
- `+<offset>;` is only added if it is different from 1.
- the last semi colon is mandatory.

the unitNames, symbols etc are available in the units.xml file in your C:\Niagara\Niagara-xxx\defaults folder

There are also keys of the TimeZone type but they are far too complicated to be done by hand.

Finally if you want to use more than one key, you can separate your key value pairs with the pipe "|" character. For example: `min=f:0.0|max=f:10.0`