

Personalize Messages Payload

The advanced configuration of the connectors allows you to personalize the payload of messages being sent from Niagara to 3rd parties and to adapt the connector to recognize the command to apply locally.

The following sections are common to all connectors

System Variables

If you open the AdvancedConfig of a connector you will notice a series of configuration about the messages destination, this is relative to each connector and its targeted environment, we will not focus on that here.



























If you have a look below, you will find a list of variables: \$(timestamp), \$(pointName)... They are listed here to indicate you they exist and you can use them, they are system variables.













Each variable represents a piece of data you can use in the payload, their name talks for themselves. But maybe a couple of explanations would help:

- **timestamp** will be the date the message is being sent following the ISO 8601 format.
- **pointName** is the name not the displayName of the point.
- **pointId**: is the id given by the InfoSource contained in the point. It's a unique identifier you can share with others. If necessary that id can be fixed using a [Strategy](#) or using [btibExcel](#)
- **pointTags** is a key value pairs separated by comma representing all the tags (direct and implied) of a point.
- **pointStatus** is the status of the point under the format {status} @{{input}}
- **pointValue** is the direct value meaning that a boolean will be sent as true/false and not by its falseText and trueText facets. For enumeration, the ordinal value will be sent. However the pointTags contains the information to build the ordinal.
- **deviceId** is the respective container unique identifier the point is being related to. It could be a Niagara device you added the DeviceExt or a PointFolder or a Node etc.
- **deviceName** and **deviceStatus** talks for themselves.

Property Sheet

Advanced Config (Advanced Config)

	Device Tags Destination	Devices Collection 
	Point Tags Destination	Points Collection 
	Point Status Destination	Points Collection 
	Point Value Destination	Points Collection 
	Reference Destination	References Collection 
	Alarm Destination	Collection Per Recipient 
	Default Variables	<div>\$ (timestamp) \$ (pointName) \$ (pointId) \$ (pointTags) \$ (pointStatus) \$ (pointValue) \$ (deviceId) \$ (deviceName) \$ (deviceStatus)</div>
	Custom Variables	<div> \$ (<input type="text"/>) <input type="text"/></div>
	Point Status Message Template	<pre>{ "pointId": "\$ (pointId)", "timestamp": "\$ (timestamp)", "value": \$ (pointValue), "status": "\$ (pointStatus)" }</pre>
	Point Value Message Template	<pre>{ "pointId": "\$ (pointId)", "timestamp": "\$ (timestamp)", "value": \$ (pointValue), "status": "\$ (pointStatus)" }</pre>
	Commands Policy	Single Point Command 
	Message Type	<input type="text" value="{json('type')}"/>
	Command Set Object	<input type="text" value="{json('').escape}"/>
	Command Device Id	<input type="text" value="{json('deviceId')}"/>
	Command Point Id	<input type="text" value="{json('pointId')}"/>
	Command Action	<input type="text" value="{json('payload.action')}"/>
	Command Value	<input type="text" value="{json('payload.value')}"/>
	Command Duration	<input type="text" value="{json('payload.duration')}"/>

 startDate	<input type="text" value="{json('payload.startDate')}"/>	
 endDate	<input type="text" value="{json('payload.endDate')}"/>	
 delta	<input type="text" value="{json('payload.delta')}"/>	
 rollUp	<input type="text" value="{json('payload.rollup')}"/>	
 historyMessageVariables	<div> <div>\$({startTimestamp})</div> <div>\$({endTimeStamp})</div> <div>\$({pointId})</div> <div>\$({deviceId})</div> <div>\$({status})</div> <div>\$({value})</div> <div>\$({trendsFlags})</div> <div>\$({count})</div> <div>\$({min})</div> <div>\$({max})</div> <div>\$({avg})</div> <div>\$({sum})</div> </div>	
 historyMessageTemplate	<div> <div>{</div> <div> <div>"startTimestamp": "\${startTimestamp}",</div> <div>"endTimeStamp": "\${endTimeStamp}",</div> <div>"deviceId": "\${deviceId}",</div> <div>"pointId": "\${pointId}",</div> <div>"trendsFlags": "\${trendsFlags}",</div> <div>"status": "\${status}",</div> <div>"value" : \${value},</div> <div>"count": \${count},</div> <div>"min": \${min},</div> </div> <div>}</div> </div>	
 Invalid Value Policy	<div>Ignore Value</div>	
 Json Message Policy	<div>Integrate Json</div>	

Use the System Variables to personalize the payloads

The system variables can then be used in the payloads template. You have two payload

- Point Status Message Template: this is the one used when the point status changes
- Point Value Message Template: this is the one used when the point value changes

The payload is a JSON object where you define key/value pairs. On the left is the name of the key and on the right the system variable you set with a \$

You can personalize it by adding a new key for example

```
{
  "pointId": "${pointId}",
  "timestamp": "${timestamp}",
  "value": "${pointValue}",
  "status": "${pointStatus}",
  "deviceName": "${deviceName}",
  "deviceId": "${deviceId}"
}
```


Or to go further in the personalization.

```
{
  "version": 1,
  "timestamp": "${timestamp}",
  "points": {
    "${pointId}": {
      "present_value": ${pointValue}
    }
  }
}
```

Define your own custom variables

One of the slot available allows you to define custom variables using [SFormat](#)

- The left part is the name of your variable
- The right part is the value you want to retrieve. The SFormat will allow to get almost any value you need. The SFormat origin will be resolved from the PointExtension itself, this is why you will probably need to access the parent.

 Custom Variables

+

)

 ?
 X

You can then use it in the payload

```
{
  "pointId": "${pointId}",
  "timestamp": "${timestamp}",
  "value": "${pointValue}",
  "status": "${pointStatus}",
  "stationName": "${stationName}"
}
```

To go further you can check the full example of the UDMI format being implemented with the Google IoT Core connector: [Configure the connector to be compliant with UDMI](#).

Personalize History messages

You can send a command to retrieve historical values at any time, to personalize the payload being sent go to the advanced config and change the template.

	History Message Variables	\$(startTimestamp) \$(endTimestamp) \$(pointId) \$(deviceId) \$(status) \$(value) \$(trendsFlags) \$(count) \$(min) \$(max) \$(avg) \$(sum)
	History Message Template	<pre>{ "startTimestamp": "\${startTimestamp}", "endTimestamp": "\${endTimestamp}", "deviceId": "\${deviceId}", "pointId": "\${pointId}", "trendsFlags": "\${trendsFlags}", "status": "\${status}", "value": \${value}, "count": \${count}, "min": \${min},</pre>

you can use all or some predefined variables.

