ibsysAddOns

Documentation for the ibsysAddOns module

Version: V3.0

20th April 2026

IBSYS GmbH
Lagerplatz 6, 8400 Winterthur
Author: Levi Jetzer

General

The ibsysAddOns module contains add-ons that extend existing Niagara building blocks and modules. These can expand and improve the functionality of the existing modules.

1. Starting with version 4.2.1.2, license credits were introduced as a license unit. The number of license credits required for each component is shown in chapter Basic Rest Api Call

The Basic Rest API Call component can be used to send simple HTTP requests. This component can, for example, be used to send any authentication requests required by the Rest Api Recipient component.

1.1. Slots

Name	Description	Format
Address	IP address or hostname of the API endpoint	Ord
Path	Path of the API endpoint	String
Http Request Method	The request method used for the call	HttpMethod
Timeout	Timeout for the API call	RelTime
Send Request On Body Change	If set to true, a request is sent whenever the body is modified	Boolean
Request Headers	Static or linked headers that are appended when the request is sent; see also Request Headers	BasicRequestHeaders
Request Params	Static or linked headers that are appended when the request is sent; see also Request Params	BasicRequestParams
Request Body	Body attached to the request	String
Response Code	Response code of the response	Integer
Response Message	Response message	String
Response Body	Response body	String
Perform Request	Sends an HTTP request	Action

1.1. Request Headers

You can add a static or linked header by right-clicking and selecting ‘Add Header’. It is called ‘linked’ because it is possible to set this value externally.

1.2. Request Params

You can add a static or linked parameter by right-clicking and selecting ‘Add Param’. It is called ‘linked’ because it is possible to set this value externally.

2. Rest Api Recipient

The REST API Recipient component is used to send alerts via HTTP REST requests. The alert data can be sent to an API endpoint either as headers, parameters or, with the help of additional components, as the body of the request.

Note: Authentication must take place outside the component, e.g. using another REST API recipient. The API key can then be extracted from the response using other components and included in the calls as a parameter or in the header.

2.1. Slots

Name	Description	Format
Address	IP address or hostname of the API endpoint	Ord
Path	Path of the API endpoint	String
Http Request Method	The request method used for the call	HttpMethod
Timeout	Timeout for the API call	RelTime
Send Request On Alarm	If set to true, a request is sent when an alarm is triggered	Boolean
Send Request On Body Change	If set to true, a request is sent whenever the body is modified	Boolean
Request Headers	Static or linked headers that are appended when the request is sent; see also Request HeadersRequest Headers	BasicRequestHeaders
Alarm Request Headers	See also Alarm Request Headers	AlarmPropKeyValuePairs
Request Params	Static or linked headers that are appended when the request is sent; see also Request Params	BasicRequestParams
Alarm Request Params	See also Alarm Request Params	AlarmPropKeyValuePairs
Request Body	Body attached to the request	String
Response Code	Response code of the response	Integer
Response Message	Response message	String
Response Body	Response body	String
Route Alarm	Forwards an alarm and sends it if necessary	Action(AlarmRecord)

2.1. Request Headers

You can add a static or linked header by right-clicking and selecting ‘Add Header’. It is called ‘linked’ because it is possible to set this value externally.

2.2. Alarm Request Headers

This property allows alarm properties to be added as a header. If ‘Static Value’ is selected, a static value can also be specified. However, this cannot be overridden externally. If left blank, this header will not be added to the call.

2.3. Request Params

You can add a static or linked parameter by right-clicking and selecting ‘Add Param’. It is called ‘linked’ because it is possible to set this value externally.

2.4. Alarm Request Params

This property allows alarm properties to be added as parameters. If ‘Static Value’ is selected, a static value can also be specified. However, this cannot be set externally. If left blank, this parameter will not be added to the call.

License Credits.

Compatibility

The ibsysAddOns module can be used from Niagara version N4.10.

Version

This documentation applies from module version 4.3.1.2.

Contact

If you have any questions, comments, suggestions or error messages, please contact our technical support:

support@ibsys.ch

www.ibsys.ch

Version index

Version	Date	Comment	Author
V1.0	10/03/2025	Created	Levi Jetzer
V2.0	12/01/2026	Edited to include the added SMS components, license credits	Levi Jetzer
V2.1	05/03/2026	License credits edited	Levi Jetzer
V3.0	20/04/2026	Edited to include the added Email and Rest API component	Levi Jetzer

Table of contents

3. Email Recipient

The Email Recipient is a component for sending emails. It allows the email addresses of existing users in the User Service to be used. Furthermore, the Email Recipient can be used to send files as attachments, such as those generated by the History CSV Exporter from the ibsysComponents module.

Like existing Email Recipients, the Email Recipient relies on the Email Service and an outgoing account.

3.1. Slots

Name	Description	Format
Enabled	Switching the component on/off	Boolean
To	'To' field of the email	UsernameEmailAddressList
Cc	'Cc' field of the email	UsernameEmailAddressList
Bcc	'Bcc' field of the email	UsernameEmailAddressList
Language	Language used when resolving format slots	String
Email Account	Outgoing account used for sending emails	OutgoingAccount
Subject	Subject of the email	Format
Body	Body of the email	Format
File Name Prepend	Prepend text to the filenames of attachments	String
File Type	File extension of the attachments (e.g. .txt, .csv, etc.)	String
Files Stored	Number of files stored	Integer
Send	Triggers the sending of the email	Action
Route File	Attach the incoming file to the email and send it	Action(Blob)
Store File	Stores the incoming file and sends all saved files the next time the "Send" command is issued	Action(Blob)

Note: The files stored in the component are only saved at runtime. This means that Java heap space is required and occupied for this purpose. Therefore, the number of stored files should be kept to a minimum. The function is designed to enable all reports to be sent in a single batch. To this end, the email containing the files should be sent shortly after generation.

It should also be noted that the maximum size of permitted attachments must not be exceeded.

4. SNMP Recipient

The SNMP Recipient is a component for sending alarms via SNMP. Alarms sent via SNMP are called traps.

A special feature of the SNMP recipient in this module is that you can configure the mapping between OID and alarm property yourself. This is usually specified by the manufacturer.

Under the “Oid Alarm Config” slot, you can create your own configuration between OID and alarm property . Any number of configuration entries can be added for configuration. This is done using the “+” in the upper left corner.

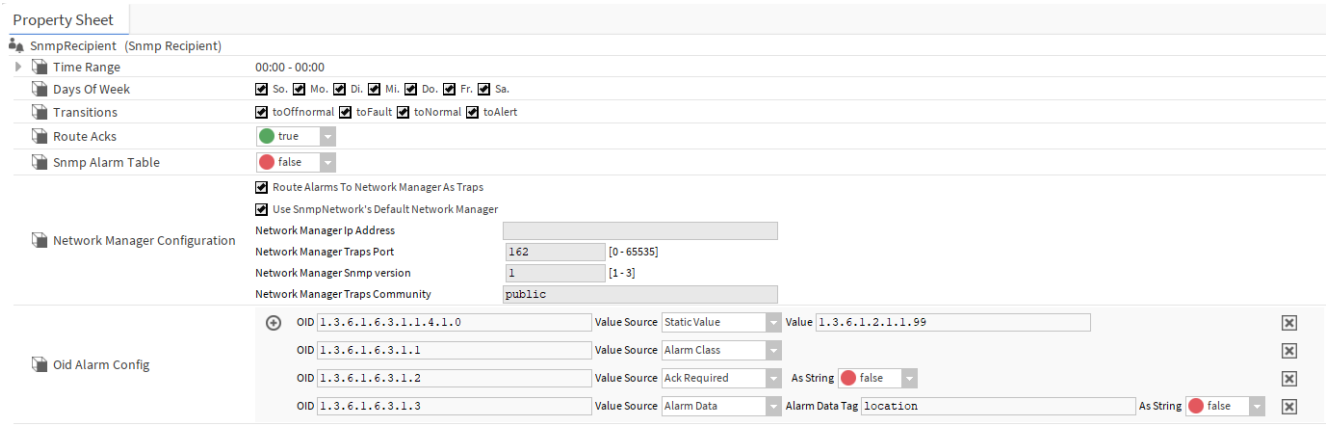
Such a configuration is based on the following properties:

OID: The OIDs are standardized or specified by the recipient. However, the following OID must be sent as well, as this is the main ID:
1.3.6.1.6.3.1.1.4.1.0

Value Source: Selection of the value source from static value or alarm properties.

Value: If “Static Value” or “Alarm Data” was selected as the value source, a static value or an alarm data tag can be entered here. For more information, see the chapter on alarm data.

As String: Since SNMP only accepts string or integer values, certain values can be transmitted either as the string “true” or as the integer “false.” If the value is not an integer, it is automatically sent as a string.



Detailed information on OIDs can be found at the following links:

https://www.alvestrand.no/objectid/top.html

https://www.rfc-editor.org/rfc/rfc1213.txt

A description of frequently requested OIDs can be found here:

https://www.alvestrand.no/objectid/1.3.6.1.2.1.1.html

4.1. Alarm Data

4.1.1. Existing fields

The existing fields under “Alarm Data” can be acquired with the following tags. To do this, simply enter the tag in the “Alarm Data Tag” field. Please note that the spelling must be exact! The following tags are available and attract the corresponding value:

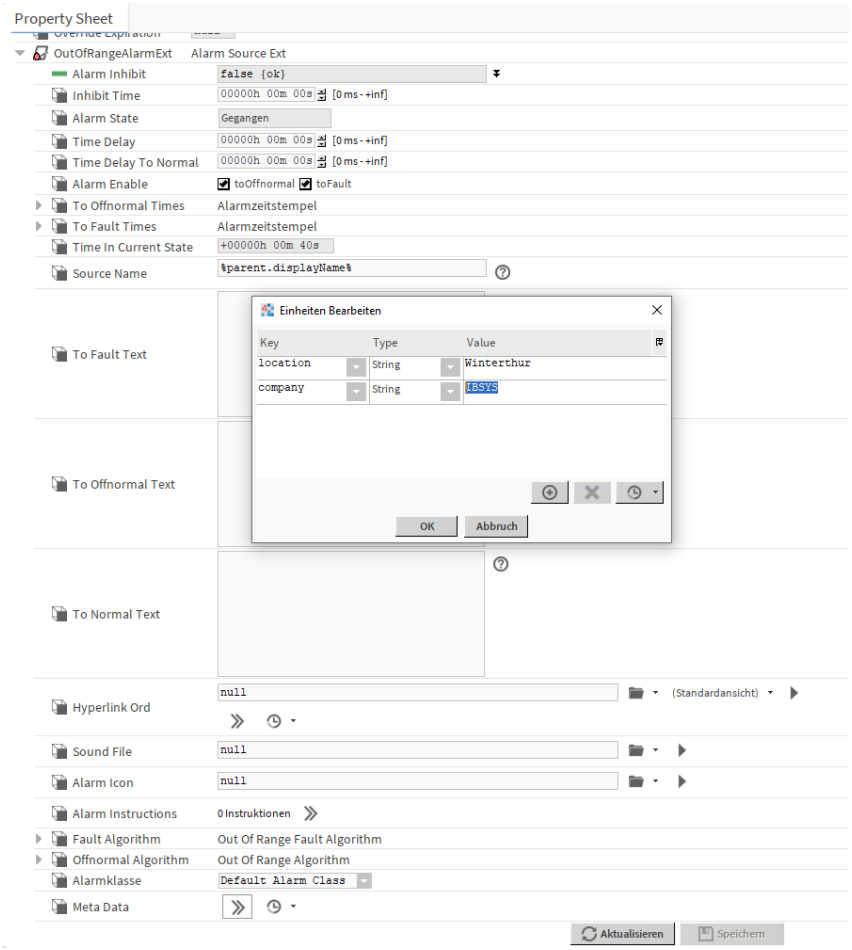
Not every alarm contains all tags in Alarm Data. If a tag does not exist for an alarm, it is not added to the SNMP trap.

Alarm Data Value	Alarm Data Tag
Present Value	presentValue
High Limit	highLimit
Low Limit	lowLimit
Message Text	msgText
Alarm Value	alarmValue
Escalated	escalated
Error Limit	errorLimit
Hyperlink Ord	hyperlinkOrd
Instructions	instructions
Controlled Value	controlledValue
New Value	newValue
Notes	notes
Notify Type	notifyType
Numeric Value	numericValue
Feedback Numeric	feedbackNumeric
Feedback Value	feedbackValue
Setpoint Value	setptValue
Status	status
Deadband	deadband
Alarm Value	alarmValue
Offnormal Value	offnormalValue
Fault Value	faultValue
Counter	Count
Time Zone	TimeZone
Source Name	sourceName
From State	fromState
To State	toState

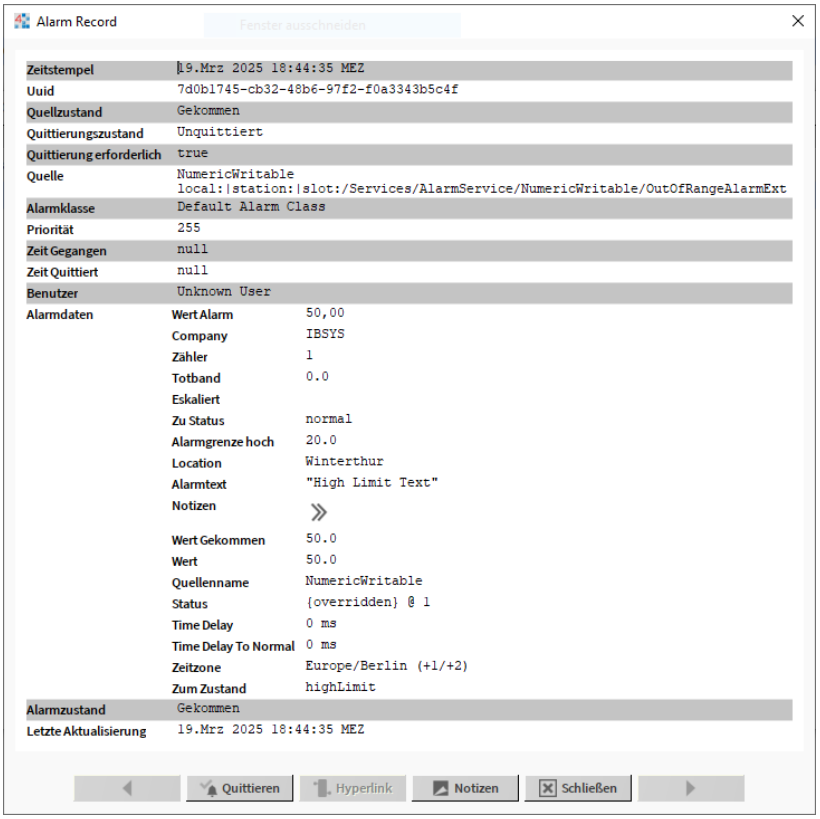
4.1.2. Additional fields

Additional metadata can also be added to the alarm, which can then be sent. This is done in the alarm extension in the “Meta Data” slot. Whatever is added there as a “key” is included with the alarm under “Alarm Data.” The value after the “key” can be retrieved from the alarm as an “Alarm Data Tag.”

Here's an example in an “OutOfRangeAlarmExt” on a NumericWritable:



The alarm configured above then appears as follows in the console:



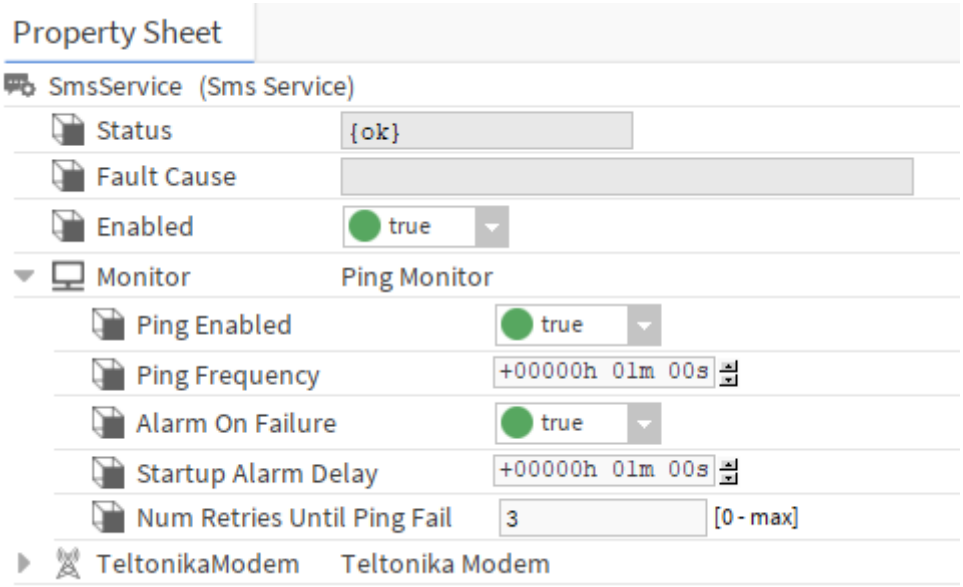
To reacquire the Mata Data Tags in SNMP Recipient, the following Alarm Data Tags must therefore be added:

Alarm Data Tag:	location	Returns value:	Winterthur
Alarm Data Tag:	company	Returns value:	IBSYS

5. SMS

5.1. Sms Service

The SMS service serves as the basic component for SMS components. This component is responsible for pinging the underlying Sms Modems.



5.2. Sms Modem

The Sms Modem component serves as the basis for integrating various modem types. This serves as preparation for later implementations.

5.2.1. Slots

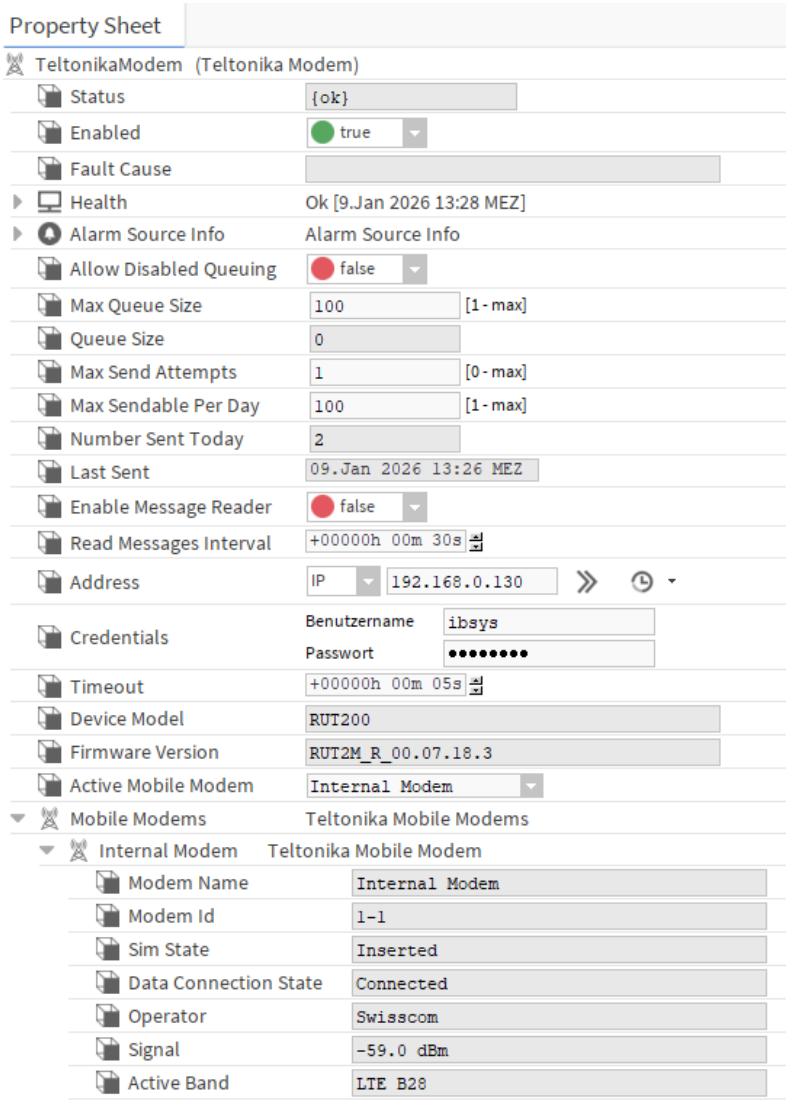
Name	Description	Format
Allow Disabled Queuing	Allows queuing when switched off	Boolean
Max Queue Size	Maximum size of the queue	Integer
Queue Size	Actual size of the queue	Integer
Max Send Attempts	Maximum number of send attempts	Integer
Max Sendable Per Day	Maximum number of SMS messages per day	Integer
Number Sent Today	Number of SMS messages sent today	Integer
Last Sent	Time of last sent	AbsTime
Ping	Ping	Action
Send	Sending an SMS manually	Action
Clear Queue	Clear Queue	Action
Process Queue	Manual processing of the queue	Action
Reset Number Sent Today	Reset the number of SMS messages sent today	Action
Read Messages	Manual reading of SMS messages	Action

5.2.2. Teltonika Modem

The Teltonika Modem represents a modem from Teltonika that can be used for handling SMS messages. This type of modem enables the sending and receiving of SMS messages.

1.1.1.1. Slots

Name	Description	Format
Address	IP address or host name of the modem	Ord
Credentials	Modem username and password	UsernameAndPassword
Timeout	Timeout for requests	RelTime
Enable Message Reader	Turning the SMS reader on/off	Boolean
Read Messages Interval	Interval for reading SMS messages	RelTime
Device Model	Device Model	String
Firmware Version	Firmware Version	String
Active Mobile Modem	Internal modem used for sending and receiving	String
Mobile Modems	Available internal modems	TeltonikaMobileModems



5.3. Sms Recipient

The SMS Recipient component can be used to send texts from alarms. The message can use the BFormat properties of an alarm in the text.

Name	Description	Format
Sms Modem	Modem used to send	String
Recipients	Recipients of SMS messages	SmsRecipientStructure
Message	Message content	BFormat

5.3.1. Sms Modem

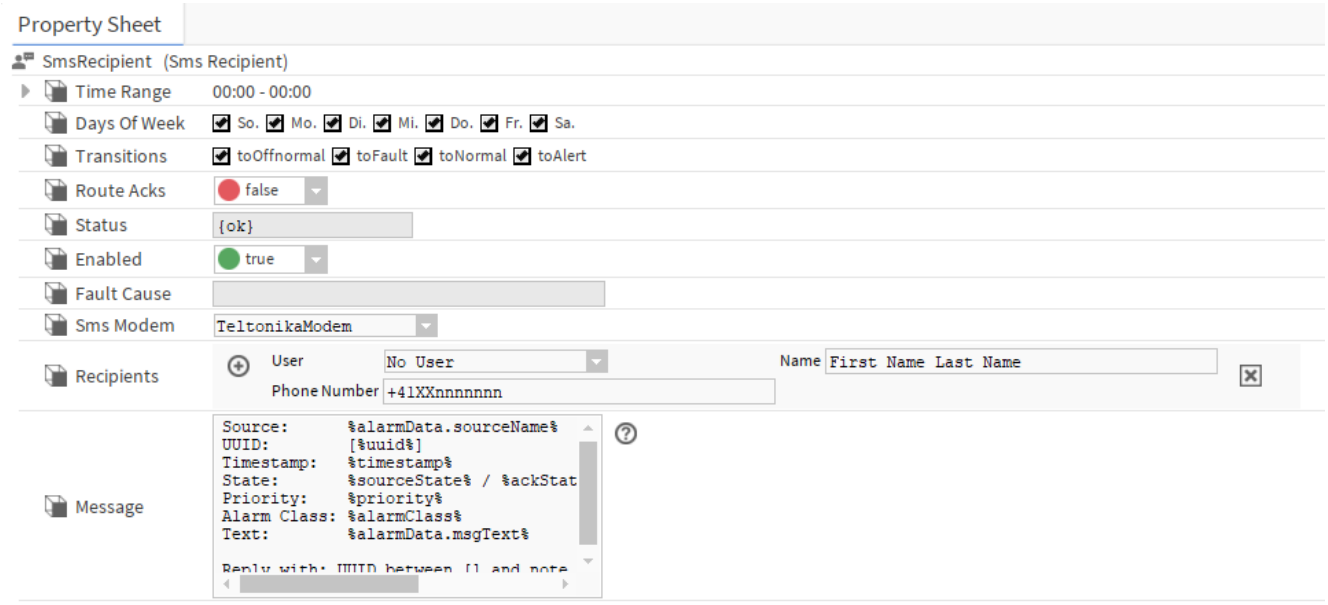
Selection from the Sms Service available modems.

5.3.2. Recipients

Select from “No User” and all users available in User Service. If a number is entered in the user, it is updated at runtime and used for sending. If you want to send to a number that is not assigned to a user, you can use “No User” and enter any number.

5.3.3. Message

The message can be entered freely and include alarm properties. These can be referenced from the alarm via BFormat. The available BFormats are described in section **Fehler! Verweisquelle konnte nicht gefunden werden.**, among other places.



5.4. Sms Alarm Acknowledger

The SMS Alarm Acknowledger module can only be inserted under an Sms Modem. To ensure that the alarm acknowledgment function works properly, the message reader must be switched on.

Name	Description	Format
Default User	For acknowledging users used if no user was found based on the number	String
Check User Permissions	Enabling/disabling user permissions check	Boolean

5.4.1. Content of the incoming message

To acknowledge an alarm via SMS, the alarm UUID must be enclosed in square brackets. Optionally, a note can be added to the acknowledgment. This must be enclosed in curly brackets.

An incoming message could look like this, for example:

[UUID], {note for alarm acknowledgment}

The outgoing alarm texts are already preconfigured so that they can be copied and used as a response to the alarm. If an alarm message is copied and sent back, the corresponding alarm will be acknowledged if necessary.

5.4.2. Default User

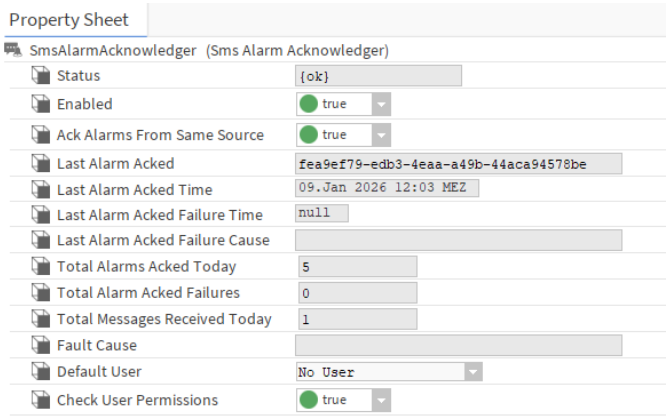
The system attempts to find a user in the User Service based on the number of the incoming message. If no user is found for the number, the selected default user is used for acknowledgment.

If “No User” is selected as the default user, acknowledgment is performed with the user name “System Default User”.

5.4.3. Check User Permissions

If this flag is set to true, the user's authorization is checked before acknowledgment. If the user does not have authorization to acknowledge, the message is discarded. If the user has the necessary rights, the alarm is acknowledged accordingly.

If the default user is used, as described in section 5.4.2, the authorization check is not performed.



5.5. Compatible Modems (Hardware)

The driver has been tested by IBSYS with the following modems.

Manufacturer	Type	Firmware Version
Teltonika	RUT200	7.18.3

5.5.1. Teltonika

You can check which Teltonika modems are compatible on their website:

https://developers.teltonika-networks.com

If in doubt, IBSYS will be happy to provide information about compatibility.

6. Basic Rest Api Call

The Basic Rest API Call component can be used to send simple HTTP requests. This component can, for example, be used to send any authentication requests required by the Rest Api Recipient component.

6.1. Slots

Name	Description	Format
Address	IP address or hostname of the API endpoint	Ord
Path	Path of the API endpoint	String
Http Request Method	The request method used for the call	HttpMethod
Timeout	Timeout for the API call	RelTime
Send Request On Body Change	If set to true, a request is sent whenever the body is modified	Boolean
Request Headers	Static or linked headers that are appended when the request is sent; see also Request Headers	BasicRequestHeaders
Request Params	Static or linked headers that are appended when the request is sent; see also Request Params	BasicRequestParams
Request Body	Body attached to the request	String
Response Code	Response code of the response	Integer
Response Message	Response message	String
Response Body	Response body	String
Perform Request	Sends an HTTP request	Action

6.2. Request Headers

You can add a static or linked header by right-clicking and selecting ‘Add Header’. It is called ‘linked’ because it is possible to set this value externally.

6.3. Request Params

You can add a static or linked parameter by right-clicking and selecting ‘Add Param’. It is called ‘linked’ because it is possible to set this value externally.

7. Rest Api Recipient

The REST API Recipient component is used to send alerts via HTTP REST requests. The alert data can be sent to an API endpoint either as headers, parameters or, with the help of additional components, as the body of the request.

Note: Authentication must take place outside the component, e.g. using another REST API recipient. The API key can then be extracted from the response using other components and included in the calls as a parameter or in the header.

7.1. Slots

Name	Description	Format
Address	IP address or hostname of the API endpoint	Ord
Path	Path of the API endpoint	String
Http Request Method	The request method used for the call	HttpMethod
Timeout	Timeout for the API call	RelTime
Send Request On Alarm	If set to true, a request is sent when an alarm is triggered	Boolean
Send Request On Body Change	If set to true, a request is sent whenever the body is modified	Boolean
Request Headers	Static or linked headers that are appended when the request is sent; see also Request HeadersRequest Headers	BasicRequestHeaders
Alarm Request Headers	See also Alarm Request Headers	AlarmPropKeyValuePairs
Request Params	Static or linked headers that are appended when the request is sent; see also Request Params	BasicRequestParams
Alarm Request Params	See also Alarm Request Params	AlarmPropKeyValuePairs
Request Body	Body attached to the request	String
Response Code	Response code of the response	Integer
Response Message	Response message	String
Response Body	Response body	String
Route Alarm	Forwards an alarm and sends it if necessary	Action(AlarmRecord)

7.2. Request Headers

You can add a static or linked header by right-clicking and selecting ‘Add Header’. It is called ‘linked’ because it is possible to set this value externally.

7.3. Alarm Request Headers

This property allows alarm properties to be added as a header. If ‘Static Value’ is selected, a static value can also be specified. However, this cannot be overridden externally. If left blank, this header will not be added to the call.

7.4. Request Params

You can add a static or linked parameter by right-clicking and selecting ‘Add Param’. It is called ‘linked’ because it is possible to set this value externally.

7.5. Alarm Request Params

This property allows alarm properties to be added as parameters. If ‘Static Value’ is selected, a static value can also be specified. However, this cannot be set externally. If left blank, this parameter will not be added to the call.

8. License Credits

Component	Required License Credits (per instance)
Email Recipient	2,000

9. Email Recipient

The Email Recipient is a component for sending emails. It allows the email addresses of existing users in the User Service to be used. Furthermore, the Email Recipient can be used to send files as attachments, such as those generated by the History CSV Exporter from the ibsysComponents module. Like existing Email Recipients, the Email Recipient relies on the Email Service and an outgoing account.

9.1. Slots

Name	Description	Format
Enabled	Switching the component on/off	Boolean
To	'To' field of the email	UsernameEmailAddressList
Cc	'Cc' field of the email	UsernameEmailAddressList
Bcc	'Bcc' field of the email	UsernameEmailAddressList
Language	Language used when resolving format slots	String
Email Account	Outgoing account used for sending emails	OutgoingAccount
Subject	Subject of the email	Format
Body	Body of the email	Format
File Name Prepend	Prepend text to the filenames of attachments	String
File Type	File extension of the attachments (e.g. .txt, .csv, etc.)	String
Files Stored	Number of files stored	Integer
Send	Triggers the sending of the email	Action
Route File	Attach the incoming file to the email and send it	Action(Blob)
Store File	Stores the incoming file	Action(Blob)

10,000

and sends all saved files the next time the “Send” command is issued	

Note: The files stored in the component are only saved at runtime. This means that Java heap space is required and occupied for this purpose. Therefore, the number of stored files should be kept to a minimum. The function is designed to enable all reports to be sent in a single batch. To this end, the email containing the files should be sent shortly after generation.
It should also be noted that the maximum size of permitted attachments must not be exceeded.

SNMP Recipient	
Sms Modem	10,000
Sms Alarm Acknowledger	10,000
Basic Rest Api Call	2,000
Rest Api Recipient	10,000