



ibsysAddOns

Dokumentation zum Modul ibsysAddOns

Version: V3.0

20. April 2026

IBSYS GmbH
Lagerplatz 6, 8400 Winterthur
Autor: Levi Jetzer

Allgemein

Im Modul ibsysAddOns sind Add Ons zu finden, welche bestehende Bausteine und Module von Niagara erweitern. Diese können den Funktionsumfang der bestehenden Module erweitern und verbessern.

1. Ab Version 4.2.1.2 wurden License Credits als Lizenzeneinheit eingeführt. Welche Komponente wie viele License Credits benötigt, wird im Kapitel Basic Rest Api Call

Mithilfe der Komponente Basic Rest Api Call können einfache HTTP-Requests abgesetzt werden. Diese Komponente kann z.B. dafür genutzt werden, um allfällige Authentication-Calls abzusetzen, welche für die Komponente Rest Api Recipient benötigt werden.

1.1. Slots

Name	Beschreibung	Format
Address	IP-Adresse oder Hostname des API-Endpunkts	Ord
Path	Pfad des API-Endpunkts	String
Http Request Method	Für den Call genutzte Request Methode	HttpMethod
Timeout	Timeout für den API-Call	RelTime
Send Request On Body Change	Wenn true wird ein Request bei Änderung des Body versendet	Boolean
Request Headers	Statische oder verknüpfte Headers, welche beim versenden des Requests angefügt werden; Siehe auch Request Headers	BasicRequestHeaders
Request Params	Statische oder verknüpfte Headers, welche beim versenden des Requests angefügt werden; Siehe auch Request Params	BasicRequestParams
Request Body	Body welcher dem Request angefügt wird	String
Response Code	Response Code der Response	Integer
Response Message	Response Message der Response	String
Response Body	Body der Response	String
Perform Request	Setzt einen HTTP-Request ab	Action

1.1. Request Headers

Mit einem rechtsklick und der Aktion «Add Header» kann ein statischer oder verknüpfter Header hinzugefügt werden. Verknüpft daher, da es möglich ist diesen Wert von aussen zu beschreiben.

1.2. Request Params

Mit einem rechtsklick und der Aktion «Add Param» kann ein statischer oder verknüpfter Parameter hinzugefügt werden. Verknüpft daher, da es möglich ist diesen Wert von aussen zu beschreiben.

2. Rest Api Recipient

Die Rest Api Recipient Komponente dient dem Alarmversand via HTTP Rest Befehl. Dabei können die Daten des Alarms wahlweise als Header, Parameter oder mithilfe zusätzlicher Komponenten als Body an einen API-Endpoint gesendet werden.

Hinweis: Eine Authentication muss ausserhalb der Komponente erfolgen, z.B. mit einem weiteren Rest API Recipient. Der API Key kann dann mithilfe anderer Komponenten aus der Response extrahiert und den Calls als Parameter oder im Header mitgegeben werden.

2.1. Slots

Name	Beschreibung	Format
Address	IP-Adresse oder Hostname des API-Endpunkts	Ord
Path	Pfad des API-Endpunkts	String
Http Request Method	Für den Call genutzte Request Methode	HttpMethod
Timeout	Timeout für den API-Call	RelTime
Send Request On Alarm	Wenn true wird ein Request beim auftreten eines Alarms versendet	Boolean
Send Request On Body Change	Wenn true wird ein Request bei Änderung des Body versendet	Boolean
Request Headers	Statische oder verknüpfte Headers, welche beim versenden des Requests angefügt werden; Siehe auch Request Headers	BasicRequestHeaders
Alarm Request Headers	Siehe auch Alarm Request Headers	AlarmPropKeyValuePairs
Request Params	Statische oder verknüpfte Headers, welche beim versenden des Requests angefügt werden; Siehe auch Request Params	BasicRequestParams
Alarm Request Params	Siehe auch Alarm Request Params	AlarmPropKeyValuePairs
Request Body	Body welcher dem Request angefügt wird	String
Response Code	Response Code der Response	Integer
Response Message	Response Message der Response	String
Response Body	Body der Response	String
Route Alarm	Leitet einen Alarm weiter und versendet diesen ggf.	Action(AlarmRecord)

2.2. Request Headers

Mit einem Rechtsklick und der Aktion «Add Header» kann ein statischer oder verknüpfter Header hinzugefügt werden. Verknüpft daher, da es möglich ist diesen Wert von aussen zu beschreiben.

2.3. Alarm Request Headers

Mithilfe dieser Property können Alarmeigenschaften als Header hinzugefügt werden. Wird «Static Value» ausgewählt, kann auch ein statischer Wert mitgegeben werden. Dieser kann jedoch nicht von aussen beschrieben werden. Wird dieser leer gelassen, wird dieser Header dem Call nicht hinzugefügt.

2.4. Request Params

Mit einem Rechtsklick und der Aktion «Add Param» kann ein statischer oder verknüpfter Parameter hinzugefügt werden. Verknüpft daher, da es möglich ist diesen Wert von aussen zu beschreiben.

2.5. Alarm Request Params

Mithilfe dieser Property können Alarmeigenschaften als Parameter hinzugefügt werden. Wird «Static Value» ausgewählt, kann auch ein statischer Wert mitgegeben werden. Dieser kann jedoch nicht von aussen beschrieben werden. Wird dieser leer gelassen, wird dieser Parameter dem Call nicht hinzugefügt.

License Credits dargestellt.

Kompatibilität

Das Modul ibsysAddOns ist ab der Niagara Version N4.10 einsetzbar.

Version

Diese Dokumentation gilt ab der Modulversion 4.3.1.2.

Kontakt

Bei Fragen, Anmerkungen, Anregungen oder Fehlermeldungen kontaktieren Sie bitte unseren technischen Support:

support@ibsys.ch

www.ibsys.ch

Versionsindex

Version	Datum	Bemerkung	Autor
V1.0	10.03.2025	Erstellung	Levi Jetzer
V2.0	12.01.2026	Bearbeitet auf die hinzugefügten SMS Komponenten, License Credits	Levi Jetzer
V2.1	05.03.2026	License Credits bearbeitet	Levi Jetzer
V3.0	20.04.2026	Bearbeitet auf die hinzugefügten Email & Rest API Komponenten	Levi Jetzer

Inhaltsverzeichnis

1. Email Recipient	3
1.1. Slots	3
2. SNMP Recipient	4
2.1. Alarm Data	5
2.1.1. Bestehende Felder	5
2.1.2. Zusätzliche Felder	6
3. SMS	8
3.1. Sms Service	8
3.2. Sms Modem	8
3.2.1. Slots	8
3.2.2. Teltonika Modem	9
3.3. Sms Recipient	10
3.3.1. Sms Modem	10
3.3.2. Recipients	10
3.3.3. Message	10
3.4. Sms Alarm Acknowledger	11
3.4.1. Inhalt der eingehenden Nachricht	11
3.4.2. Default User	11
3.4.3. Check User Permissions	11
3.5. Kompatible Modems (Hardware)	12
3.5.1. Teltonika	12
4. Basic Rest Api Call	13
4.1. Slots	13
4.1. Request Headers	13
4.2. Request Params	13
5. Rest Api Recipient	14
5.1. Slots	14
5.2. Request Headers	15
5.3. Alarm Request Headers	15
5.4. Request Params	15
5.5. Alarm Request Params	15
6. License Credits	16

3. Email Recipient

Der Email Recipient ist ein Komponente zum Versand von Emails. Dabei können die Email-Adressen der bestehenden Benutzer im User Service verwendet werden. Weiter können mit dem Email-Recipient Dateien als Anhang versendet werden, welche z.B. durch den History CSV Exporter aus dem Modul ibsysComponents generiert werden.

Der Email Recipient ist wie auch bestehende Email Recipient auf den Email Service sowie einen Outgoing Account angewiesen.

3.1. Slots

Name	Beschreibung	Format
Enabled	Ein-/Ausschalten der Komponente	Boolean
To	«An» Feld des Emails	UsernameEmailAddressList
Cc	«Cc» Feld des Emails	UsernameEmailAddressList
Bcc	«Bcc» Feld des Emails	UsernameEmailAddressList
Language	Sprache bei der Auflösung der Format-Slots verwendet wird	String
Email Account	Outgoing Account, welcher für den Versand verwendet wird	OutgoingAccount
Subject	Betreff des Emails	Format
Body	Nachricht des Emails	Format
File Name Prepend	Prepend-Text für den Dateinamen der Anhänge	String
File Type	Dateiendung der Anhänge (z.B.: .txt, .csv, usw.)	String
Files Stored	Anzahl gespeicherter Files	Integer
Send	Löst den Versand des Emails aus	Action
Route File	Hängt die eingehende Datei an das Email und versendet dieses	Action(Blob)
Store File	Speichert die eingehende Datei und versendet alle gespeicherten Dateien beim nächsten Send Befehl	Action(Blob)

Hinweis: Die gespeicherten Files in der Komponente werden nur zur Laufzeit gespeichert. Das bedeutet, das dafür Java-Heap benötigt bzw. belegt wird. Damit sollten die gespeicherten Files in Grenzen gehalten werden. Die Funktion ist dafür gedacht, alle Reports gesammelt versenden zu können. Dafür sollte kurz nach der Generierung das Mail mit den Files versandt werden.

Weiter ist zu beachten, dass die maximale Grösse erlaubter Anhänge nicht überschritten wird.

4. SNMP Recipient

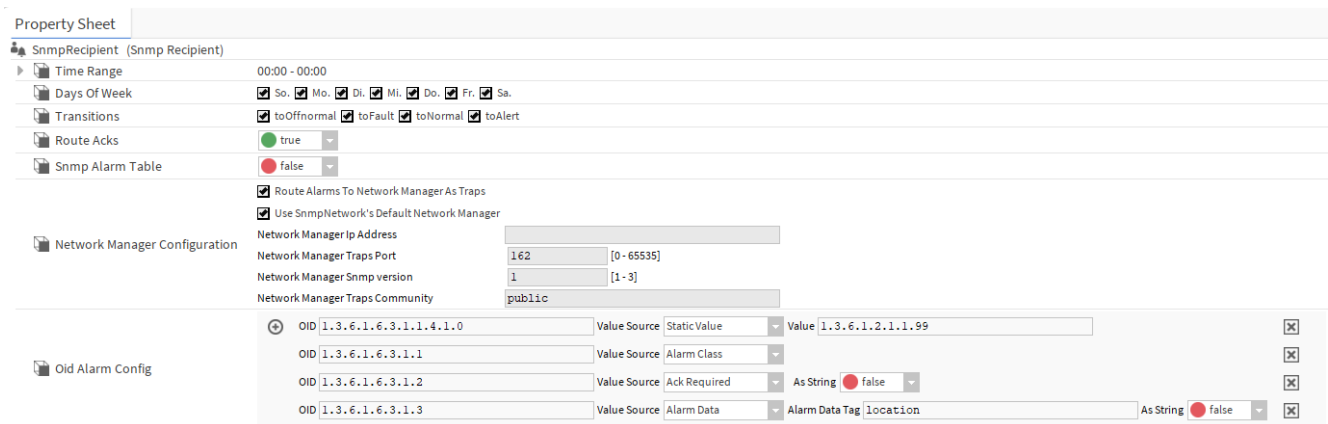
Der SNMP Recipient ist ein Komponente zum Versand von Alarmen via SNMP. Alarme welche per SNMP versendet werden, nennt man Traps.

Beim SNMP Recipient in diesem Modul ist speziell, dass man das Mapping zwischen OID und Alarm-Property selbst konfigurieren kann. Meist ist dies vom Hersteller vorgegeben.

Unter dem Slot «Oid Alarm Config» kann die eigene Konfiguration zwischen OID und Alarm-Property erstellt werden. Zur Konfiguration können beliebig viele Konfigurations-Einträge hinzugefügt werden. Dies geschieht mit dem «+» oben links.

Eine solche Konfiguration baut sich aus den folgenden Eigenschaften auf:

- OID:** Die OIDs sind standardisiert oder werden durch dem Empfänger vorgegeben. Folgende OID muss jedoch mitgesendet werden, da dies die Haupt-ID ist:
1.3.6.1.6.3.1.1.4.1.0
- Value Source:** Auswahl der Wertequelle aus Statischem Wert oder Alarmeigenschaften.
- Value:** Falls als Value Source "Static Value" oder "Alarm Data" ausgewählt wurde, kann hier ein statischer Wert oder ein Alarm Data Tag eingegeben werden. Mehr dazu im Kapitel Alarm Data.
- As String:** Da SNMP nur String oder Integer Werte akzeptiert, können gewisse Werte wahlweise als String «true» oder als Integer «false» übermittelt werden. Falls der Wert kein Integer sein sollte, wird er automatisch als String versendet.



OID	Value Source	Value	As String
1.3.6.1.6.3.1.1.4.1.0	Static Value	1.3.6.1.2.1.1.99	
1.3.6.1.6.3.1.1	Alarm Class		
1.3.6.1.6.3.1.2	Ack Required		false
1.3.6.1.6.3.1.3	Alarm Data	location	false

Detaillierte Informationen zu OIDs können unter folgenden Links gefunden werden:

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

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

Die Beschreibung oft geforderte OIDs kann hier gefunden werden:

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

4.1. Alarm Data

4.1.1. Bestehende Felder

Die bestehenden Felder unter «Alarm Data» können mit folgenden Tags akquiriert werden. Dazu muss lediglich der Tag in das Feld «Alard Data Tag» eingegeben werden. Es ist auf exakte Schreibweise zu achten! Folgende Tags stehen zur Verfügung und ziehen den entsprechenden Wert an:

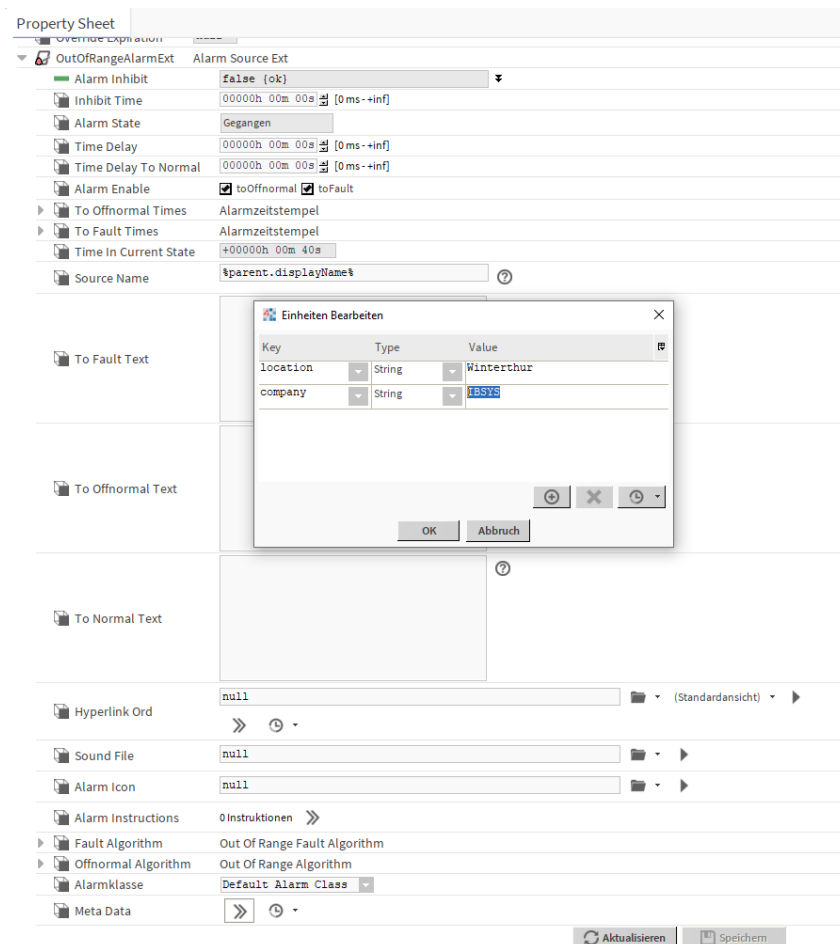
Nicht jeder Alarm enthält in Alarm Data alle Tags. Falls ein Tag bei einem Alarm nicht existiert, wird er nicht zum SNMP Trap hinzugefügt.

Alarm Data Wert	Alarm Data Tag
Aktueller Wert	presentValue
Alarmgrenze hoch	highLimit
Alarmgrenze niedrig	lowLimit
Alarmtext	msgText
Alarm Data Wert	alarmValue
Eskaliert	escalated
Grenze Fehler	errorLimit
Hyperlink Ord	hyperlinkOrd
Instruktionen	instructions
Istwert	controlledValue
Neuer Wert	newValue
Notizen	notes
Notify Typ	notifyType
Numerischer Wert	numericValue
Rückmeldung numerisch	feedbackNumeric
Rückmeldung Wert	feedbackValue
Sollwert	setptValue
Status	status
Totband	deadband
Wert Alarm	alarmValue
Wert Gekommen	offnormalValue
Wert Fehler	faultValue
Zähler	Count
Zeitzone	TimeZone
Quellenname	sourceName
Vom Zustand	fromState
Zum Zustand	toState

4.1.2. Zusätzliche Felder

Es können dem Alarm auch weitere Meta Daten hinzugefügt werden, welche dann versendet werden können. Dies geschieht in der Alarm Extension im Slot «Meta Data». Was dort als «key» hinzugefügt wird, wird dem Alarm unter «Alarm Data» mitgegeben. Der Wert hinter dem «key» kann als «Alarm Data Tag» wieder aus dem Alarm herausgeholt werden.

Dazu folgendes Beispiel in einer «OutOfRangeAlarmExt» auf einem NumericWritable:



The screenshot shows the 'Property Sheet' for an 'OutOfRangeAlarmExt' object. The 'Alarm Source Ext' section is expanded, showing various configuration fields. A dialog box titled 'Einheiten Bearbeiten' (Edit Units) is open, displaying a table with two rows of meta-data keys.

Key	Type	Value
location	String	Winterthur
company	String	IBSYS

The main property sheet includes the following visible fields:

- Alarm Inhibit: false [ok]
- Inhibit Time: 00000h 00m 00s [0 ms - +inf]
- Alarm State: Gegangen
- Time Delay: 00000h 00m 00s [0 ms - +inf]
- Time Delay To Normal: 00000h 00m 00s [0 ms - +inf]
- Alarm Enable: toOffnormal toFault
- To Offnormal Times: Alarmzeitstempel
- To Fault Times: Alarmzeitstempel
- Time In Current State: +00000h 00m 40s
- Source Name: %parent.displayName%
- To Fault Text
- To Offnormal Text
- To Normal Text
- Hyperlink Ord: null
- Sound File: null
- Alarm Icon: null
- Alarm Instructions: 0 Instruktionen
- Fault Algorithm: Out Of Range Fault Algorithm
- Offnormal Algorithm: Out Of Range Algorithm
- Alarmklasse: Default Alarm Class
- Meta Data

Buttons at the bottom right include 'Aktualisieren' and 'Speichern'.

Der oben konfigurierte Alarm sieht in der Konsole wie folgt aus:

Alarm Record
Fenster ausschneiden
✕

Zeitstempel	19.Mrz 2025 18:44:35 MEZ	
Uuid	7d0b1745-cb32-48b6-97f2-f0a3343b5c4f	
Quellzustand	Gekommen	
Quittierungszustand	Unquittiert	
Quittierungserforderlich	true	
Quelle	NumericWritable local: station: slot:/Services/AlarmService/NumericWritable/OutOfRangeAlarmExt	
Alarmklasse	Default Alarm Class	
Priorität	255	
Zeit Gegangen	null	
Zeit Quittiert	null	
Benutzer	Unknown User	
Alarmdaten	Wert Alarm	50,00
	Company	IBSYS
	Zähler	1
	Totband	0.0
	Eskaliert	
	Zu Status	normal
	Alarmgrenze hoch	20.0
	Location	Winterthur
	Alarmtext	"High Limit Text"
	Notizen	»»
	Wert Gekommen	50.0
	Wert	50.0
	Quellenname	NumericWritable
	Status	{overridden} @ 1
	Time Delay	0 ms
	Time Delay To Normal	0 ms
	Zeitzone	Europe/Berlin (+1/+2)
	Zum Zustand	highLimit
Alarmzustand	Gekommen	
Letzte Aktualisierung	19.Mrz 2025 18:44:35 MEZ	

◀
🔔 Quittieren
🔗 Hyperlink
📄 Notizen
✕ Schließen
▶

Um die Mata Data Tags in SNMP Recipient wieder zu akquirieren, müssen folglich folgende Alarm Data Tags hinzugefügt werden:

Alarm Data Tag: location Ergibt Wert: Winterthur

Alarm Data Tag: company Ergibt Wert: IBSYS

5. SMS

5.1. Sms Service

Als Basiskomponente für SMS Komponenten dient der Sms Service. Diese Komponente kümmert sich um das pinggen der darunterliegenden Sms Modems.

Property Sheet

SmsService (Sms Service)

Status

Fault Cause

Enabled true

Monitor Ping Monitor

Ping Enabled true

Ping Frequency

Alarm On Failure true

Startup Alarm Delay

Num Retries Until Ping Fail [0 - max]

TeltonikaModem Teltonika Modem

5.2. Sms Modem

Die Komponente Sms Modem dient als Basis für die Integration von verschiedenen Modem Typen. Dies dient als Vorbereitung für spätere Implementationen.

5.2.1. Slots

Name	Beschreibung	Format
Allow Disabled Queuing	Erlaubt das Queuing in ausgeschaltetem Zustand	Boolean
Max Queue Size	Maximale Grösse der Queue	Integer
Queue Size	Aktuelle Grösse der Queue	Integer
Max Send Attempts	Maximale Anzahl Sendeversuche	Integer
Max Sendable Per Day	Maximale Anzahl SMS pro Tag	Integer
Number Sent Today	Anzahl heute gesendeter SMS	Integer
Last Sent	Zeitpunkt des letzten Versands	AbsTime
Ping	Ping	Action
Send	Manuelles Versenden eines SMS	Action
Clear Queue	Queue löschen	Action
Process Queue	Manuelles Abarbeiten der Queue	Action
Reset Number Sent Today	Zurücksetzen der Anzahl heute gesendeter SMS	Action
Read Messages	Manuelles lesen von SMS	Action

5.2.2. Teltonika Modem

Die Komponente Teltonika Modem repräsentiert ein Modem von Teltonika, welches für das Handling von SMS verwendet werden kann. Ein solches Modem ermöglicht das Senden und Empfangen von SMS.

1.1.1.1. Slots

Name	Beschreibung	Format
Address	IP Adresse oder Hostname des Modems	Ord
Credentials	Benutzername und Passwort des Modems	UsernameAndPassword
Timeout	Timeout für die Anfragen	RelTime
Enable Message Reader	Ein-/Ausschalten des SMS Lesers	Boolean
Read Messages Interval	Intervall zum Lesen von SMS	RelTime
Device Model	Gerätemodell	String
Firmware Version	Firmwareversion	String
Active Mobile Modem	Zum Versand und Empfang genutztes internes Modem	String
Mobile Modems	Vorhandene interne Modems	TeltonikaMobileModems

Property Sheet

TeltonikaModem (Teltonika Modem)

- Status
- Enabled true
- Fault Cause
- Health Ok [9.Jan 2026 13:28 MEZ]
- Alarm Source Info Alarm Source Info
- Allow Disabled Queuing false
- Max Queue Size [1-max]
- Queue Size
- Max Send Attempts [0-max]
- Max Sendable Per Day [1-max]
- Number Sent Today
- Last Sent
- Enable Message Reader false
- Read Messages Interval
- Address
- Credentials
 - Benutzername
 - Passwort
- Timeout
- Device Model
- Firmware Version
- Active Mobile Modem
- Mobile Modems Teltonika Mobile Modems
 - Internal Modem Teltonika Mobile Modem
 - Modem Name
 - Modem Id
 - Sim State
 - Data Connection State
 - Operator
 - Signal
 - Active Band

5.3. Sms Recipient

Die Komponente Sms Recipient kann zum Versand von Texten aus Alarmen verwendet werden. Dabei kann die Nachricht mit BFormat Eigenschaften eines Alarms im Text verwendet werden.

Name	Beschreibung	Format
Sms Modem	Zum Versand genutztes Modem	String
Recipients	Empfänger der Sms Nachrichten	SmsRecipientStructure
Message	Inhalt der Nachricht	BFormat

5.3.1. Sms Modem

Auswahl aus den im Sms Service verfügbaren Modems.

5.3.2. Recipients

Auswahl aus «No User» und allen im User Service verfügbaren Benutzer. Falls im Benutzer eine Nummer eingetragen ist, wird diese zur Laufzeit aktualisiert und für den Versand verwendet. Möchte an eine Nummer versendet werden, welche keinem Benutzer zugeordnet ist, kann «No User» verwendet und frei eine Nummer eingetragen werden.

5.3.3. Message

Die Message kann frei eingetragen werden und Eigenschaften von Alarmen beinhalten. Diese können via BFormat aus dem Alarm verwiesen werden. Welche BFormat zur Verfügung stehen ist unter anderem unter Kapitel 4.1 beschrieben.

Property Sheet

SmsRecipient (Sms Recipient)

Time Range: 00:00 - 00:00

Days Of Week: So. Mo. Di. Mi. Do. Fr. Sa.

Transitions: toOffnormal toFault toNormal toAlert

Route Acks: false

Status: {ok}

Enabled: true

Fault Cause:

Sms Modem: TeltonikaModem

Recipients:

+	User	No User	Name	First Name Last Name	✕
	Phone Number	+41XXnnnnnnn			

Message:


```
Source:      %alarmData.sourceName%
UUID:       [%uuid%]
Timestamp:  %timestamp%
State:      %sourceState% / %ackStat
Priority:    %priority%
Alarm Class: %alarmClass%
Text:       %alarmData.msgText%

Reply with: UUID between [%] and nota
```

5.4. Sms Alarm Acknowledger

Der Baustein Sms Alarm Acknowledger kann nur unter einem Sms Modem eingefügt werden. Damit die Funktion der Alarmquittierung gewährleistet ist, muss der Message Reader eingeschaltet sein.

Name	Beschreibung	Format
Default User	Für die Quittierung verwendeter Benutzer, wenn aufgrund der Nummer kein Benutzer gefunden wurde	String
Check User Permissions	Ein-/Ausschalten der Berechtigungsprüfung des Benutzers	Boolean

5.4.1. Inhalt der eingehenden Nachricht

Um einen Alarm via SMS quittieren zu können, wird die Alarm UUID zwischen zwei eckigen Klammern erwartet. Optional kann eine Notiz zur Quittierung hinzugefügt werden. Diese wird in geschwungenen Klammern erwartet.

Eine eingehende Nachricht könnte zum Beispiel so aussehen:

[UUID], {Notiz zur Alarm Quittierung}

Die ausgehenden Alarmtexte sind bereits so vorkonfiguriert, dass dieser wiederum kopiert und als Antwort auf den Alarm verwendet werden kann. Wird also eine Alarmnachricht kopiert und zurückgesendet, wird der entsprechende Alarm ggf. quittiert.

5.4.2. Default User

Aufgrund der Nummer der eingehenden Nachricht wird versucht ein Benutzer im User Service zu finden. Wird kein Benutzer zur Nummer gefunden, wird zur Quittierung der ausgewählte Default User verwendet.

Wird als Default User «No User» ausgewählt, wird eine Quittierung mit dem Benutzernamen «System Default User» durchgeführt.

5.4.3. Check User Permissions

Ist dieses Flag auf true, wird vor der Quittierung auf die Berechtigung des Benutzers geprüft. Sollte der Benutzer keine Berechtigung zur Quittierung besitzen, wird die Nachricht verworfen. Besitzt der Benutzer die benötigten Rechte, wird der Alarm entsprechend quittiert.

Falls der Default Benutzer verwendet wird, wie in Kapitel 5.4.2 beschrieben, wird die Berechtigungsprüfung nicht durchgeführt.

Property Sheet

SmsAlarmAcknowledger (Sms Alarm Acknowledger)

Status	{ok}
Enabled	<input checked="" type="checkbox"/> true
Ack Alarms From Same Source	<input checked="" type="checkbox"/> true
Last Alarm Acked	fea9ef79-edb3-4eaa-a49b-44aca94578be
Last Alarm Acked Time	09. Jan 2026 12:03 MEZ
Last Alarm Acked Failure Time	null
Last Alarm Acked Failure Cause	
Total Alarms Acked Today	5
Total Alarm Acked Failures	0
Total Messages Received Today	1
Fault Cause	
Default User	No User
Check User Permissions	<input checked="" type="checkbox"/> true

5.5. Kompatible Modems (Hardware)

Der Treiber wurde durch IBSYS mit folgenden Modems getestet.

Hersteller	Typ	Firmware Version
Teltonika	RUT200	7.18.3

5.5.1. Teltonika

Welche Modems von Teltonika kompatibel sind, lässt sich auf deren Website prüfen:

<https://developers.teltonika-networks.com>

Im Zweifelsfall gibt IBSYS gerne Auskunft über Kompatibilitäten.

6. Basic Rest Api Call

Mithilfe der Komponente Basic Rest Api Call können einfache HTTP-Requests abgesetzt werden. Diese Komponente kann z.B. dafür genutzt werden, um allfällige Authentication-Calls abzusetzen, welche für die Komponente Rest Api Recipient benötigt werden.

6.1. Slots

Name	Beschreibung	Format
Address	IP-Adresse oder Hostname des API-Endpunkts	Ord
Path	Pfad des API-Endpunkts	String
Http Request Method	Für den Call genutzte Request Methode	HttpMethod
Timeout	Timeout für den API-Call	RelTime
Send Request On Body Change	Wenn true wird ein Request bei Änderung des Body versendet	Boolean
Request Headers	Statische oder verknüpfte Headers, welche beim versenden des Requests angefügt werden; Siehe auch Request Headers	BasicRequestHeaders
Request Params	Statische oder verknüpfte Headers, welche beim versenden des Requests angefügt werden; Siehe auch Request Params	BasicRequestParams
Request Body	Body welcher dem Request angefügt wird	String
Response Code	Response Code der Response	Integer
Response Message	Response Message der Response	String
Response Body	Body der Response	String
Perform Request	Setzt einen HTTP-Request ab	Action

6.2. Request Headers

Mit einem rechtsklick und der Aktion «Add Header» kann ein statischer oder verknüpfter Header hinzugefügt werden. Verknüpft daher, da es möglich ist diesen Wert von aussen zu beschreiben.

6.3. Request Params

Mit einem rechtsklick und der Aktion «Add Param» kann ein statischer oder verknüpfter Parameter hinzugefügt werden. Verknüpft daher, da es möglich ist diesen Wert von aussen zu beschreiben.

7. Rest Api Recipient

Die Rest Api Recipient Komponente dient dem Alarmversand via HTTP Rest Befehl. Dabei können die Daten des Alarms wahlweise als Header, Parameter oder mithilfe zusätzlicher Komponenten als Body an einen API-Endpoint gesendet werden.

Hinweis: Eine Authentication muss ausserhalb der Komponente erfolgen, z.B. mit einem weiteren Rest API Recipient. Der API Key kann dann mithilfe anderer Komponenten aus der Response extrahiert und den Calls als Parameter oder im Header mitgegeben werden.

7.1. Slots

Name	Beschreibung	Format
Address	IP-Adresse oder Hostname des API-Endpunkts	Ord
Path	Pfad des API-Endpunkts	String
Http Request Method	Für den Call genutzte Request Methode	HttpMethod
Timeout	Timeout für den API-Call	RelTime
Send Request On Alarm	Wenn true wird ein Request beim auftreten eines Alarms versendet	Boolean
Send Request On Body Change	Wenn true wird ein Request bei Änderung des Body versendet	Boolean
Request Headers	Statische oder verknüpfte Headers, welche beim versenden des Requests angefügt werden; Siehe auch Request Headers	BasicRequestHeaders
Alarm Request Headers	Siehe auch Alarm Request Headers	AlarmPropKeyValuePairs
Request Params	Statische oder verknüpfte Headers, welche beim versenden des Requests angefügt werden; Siehe auch Request Params	BasicRequestParams
Alarm Request Params	Siehe auch Alarm Request Params	AlarmPropKeyValuePairs
Request Body	Body welcher dem Request angefügt wird	String
Response Code	Response Code der Response	Integer
Response Message	Response Message der Response	String
Response Body	Body der Response	String
Route Alarm	Leitet einen Alarm weiter und versendet diesen ggf.	Action(AlarmRecord)

7.2. Request Headers

Mit einem Rechtsklick und der Aktion «Add Header» kann ein statischer oder verknüpfter Header hinzugefügt werden. Verknüpft daher, da es möglich ist diesen Wert von aussen zu beschreiben.

7.3. Alarm Request Headers

Mithilfe dieser Property können Alarmeigenschaften als Header hinzugefügt werden. Wird «Static Value» ausgewählt, kann auch ein statischer Wert mitgegeben werden. Dieser kann jedoch nicht von aussen beschrieben werden. Wird dieser leer gelassen, wird dieser Header dem Call nicht hinzugefügt.

7.4. Request Params

Mit einem Rechtsklick und der Aktion «Add Param» kann ein statischer oder verknüpfter Parameter hinzugefügt werden. Verknüpft daher, da es möglich ist diesen Wert von aussen zu beschreiben.

7.5. Alarm Request Params

Mithilfe dieser Property können Alarmeigenschaften als Parameter hinzugefügt werden. Wird «Static Value» ausgewählt, kann auch ein statischer Wert mitgegeben werden. Dieser kann jedoch nicht von aussen beschrieben werden. Wird dieser leer gelassen, wird dieser Parameter dem Call nicht hinzugefügt.

8. License Credits

Komponente	Benötigte License Credits (pro Instanz)
Email Recipient	2'000

10'000

9. Email Recipient

Der Email Recipient ist ein Komponent zum Versand von Emails. Dabei können die Email-Adressen der bestehenden Benutzer im User Service verwendet werden. Weiter können mit dem Email-Recipient Dateien als Anhang versendet werden, welche z.B. durch den History CSV Exporter aus dem Modul ibsysComponents generiert werden.

Der Email Recipient ist wie auch bestehende Email Recipient auf den Email Service sowie einen Outgoing Account angewiesen.

9.1. Slots

Name	Beschreibung	Format
Enabled	Ein-/Ausschalten der Komponente	Boolean
To	«An» Feld des Emails	UsernameEmailAddressList
Cc	«Cc» Feld des Emails	UsernameEmailAddressList
Bcc	«Bcc» Feld des Emails	UsernameEmailAddressList
Language	Sprache bei der Auflösung der Format-Slots verwendet wird	String
Email Account	Outgoing Account, welcher für den Versand verwendet wird	OutgoingAccount
Subject	Betreff des Emails	Format
Body	Nachricht des Emails	Format
File Name Prepend	Prepend-Text für den Dateinamen der Anhänge	String
File Type	Dateiendung der Anhänge (z.B.: .txt, .csv, usw.)	String
Files Stored	Anzahl gespeicherter Files	Integer
Send	Löst den Versand des Emails aus	Action

Route File	Hängt die eingehende Datei an das Email und versendet dieses	Action(Blob)	
Store File	Speichert die eingehende Datei und versendet alle gespeicherten Dateien beim nächsten Send Befehl	Action(Blob)	
Hinweis: Die gespeicherten Files in der Komponente werden nur zur Laufzeit gespeichert. Das bedeutet, das dafür Java-Heap benötigt bzw. belegt wird. Damit sollten die gespeicherten Files in Grenzen gehalten werden. Die Funktion ist dafür gedacht, alle Reports gesammelt versenden zu können. Dafür sollte kurz nach der Generierung das Mail mit den Files versandt werden. Weiter ist zu beachten, dass die maximale Grösse erlaubter Anhänge nicht überschritten wird.			
SNMP Recipient			
Sms Modem			10'000
Sms Alarm Acknowledger			10'000
Basic Rest Api Call			2'000
Rest Api Recipient			10'000