Rijkswaterstaat
Ministry of Infrastructure and Water Management

RWS INFORMATION

Winter and road service area maintenance equipment Data Exchange

Date	June 30, 2022
Status	Final
Version	1.1

Colophon

Published by	Rijkswaterstaat – Verkeer en Watermanagement
Information	A. van den Hoek / N. Stroo
Document ID	2022_WARSAME_DATA-EXCHANGE_1.1
Date	June 30, 2022
Status	Closed
Version	1.1

Contents

1 Notice

This document is a prequel to EN15430-3. The moment EN15430-3 is officially published it will replace this document with the exception of some of the appendices C, E, F and G.

2 Introduction

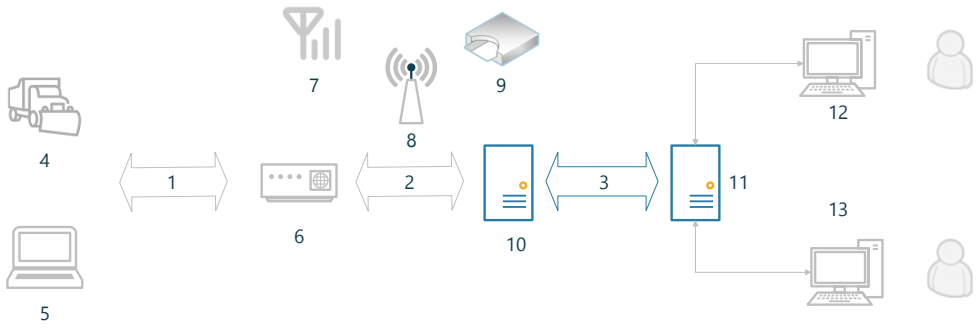
This document was commissioned by Rijkswaterstaat. This document describes the functionality with regard to data acquisition and transmission of Winter and road service area maintenance equipment.

The goal of this EN 15430 in combination with this document is to allow interoperability between systems (hardware and software) of different vendors. A customer should be able to combine:

- any on-vehicle equipment (e.g. spreaders and ploughs),
- any on-vehicle data acquisition systems (e.g. board computers or enhanced control boxes),
- any client application software (e.g. data bases, analysing or accounting software),

as long as they follow the EN 15430 standards.

EN 15430-1:2007+A1:2011 defines the on-board communication (flow 1) between on-vehicle equipment (data handler) and on-vehicle data acquisition systems (board computer). This document is meant to describe the data structure, types, ranges, protocol and initial settings required by the producer and consumer to be able to send data generated by vehicles in near real time to any third-party data client over the internet. Figure 1 represents the whole data flow chain from vehicle to office system.



Key

1	Flow 1	8	WLAN
2	Flow 2	9	M-CARD
3	Flow 3	10	Producer
4	Data handler 1	11	Consumer
5	Data handler n	12	Client application 1
6	Board computer	13	Client application n
7	GSM/GPRS		

Figure 1 — Transmission flow

Data collected is operating data of the vehicles, which contain time information, geo reference data and machine status data. This data is stored in different memories: on the vehicle, in the facility where the vehicles are maintained and in office, where data is retrieved and analysed. Due to the fact that the collected data is used not only to supervise work contents and results, but also as proof in case of accidents, the integrity and the correctness of the data is important and indispensable.

This document contains seamless integration of mechanisms to deliver data secure from a Producer to a Consumer (flow 3). It does not define any specific rules for items belonging to flow 1 or flow 2. Data transfer between Producer and Consumer must be lossless, this means that reduction of data is not allowed. Lossy compression methods are not allowed.

In Figure 1, Section 1 defines the interface between devices and board computer, as described in EN 15430-1:2007+A1:2011. Section 2 addresses the combination of data from different streams and the transmission to a generic information supplier server. Section 3 addresses the data transfer (flow 3) between the Producer and the Consumer and is the purpose of this document.

3 Scope

The function of EN 15430 is to combine any vehicle equipment with different board computers to any client application server.

The interface and protocol needed between the information supplier server and the client application server (flow 3) is the sole object of the present Technical Specification. It makes distribution of vehicle data possible without any restrictions to the technology used to gather the data like manufacturer specific protocols, WLANS systems, memory cards etc.

4 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies. Information about acquiring these documents can be found in appendix D of this document.

EN 154301:2007 + A1:2011, *Winter and road service area maintenance equipments — Data acquisition and transmission — Part 1: In vehicle data acquisition*

EN 154301 A1:2015, *Winter and road service area maintenance equipments — Data acquisition and transmission — Part 1: In vehicle data acquisition*

CEN/TS 15430-2:2012, *Winter and road service area maintenance equipments — Data acquisition and transmission — Part 2: Protocol for data transfer between information supplier and client application server*

FIELDING, Roy, et al. RFC 2616: Hypertext transfer protocol–HTTP/1.1. 1999

5 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

Producer

Entity able to distribute previously stored vehicle information to consumers using the interface and protocol described in this document.

Consumer

Entity able to retrieve the information provided by the Producer.

64-bit Integer

https://en.wikipedia.org/wiki/Integer_(computer_science)

A signed integer with a maximum value of 9223372036854775807 and a minimum value of −9223372036854775808

6 Abbreviated terms

For the purposes of this document, the following abbreviations apply.

ASCII

American national Standard Code for Information Interchange

API

Application Programming Interface

HTTP

Hyper Text Transfer Protocol

HTTPS

Hyper Text Transfer Protocol Secure

JSON

JavaScript Object Notation

REST

Representational state transfer

RFC

Request For Comments

UTC

Coordinated Universal Time

WAN

Wide Area Network

7 System overview

A system is built of one or more Producers connected to one or more Consumers.

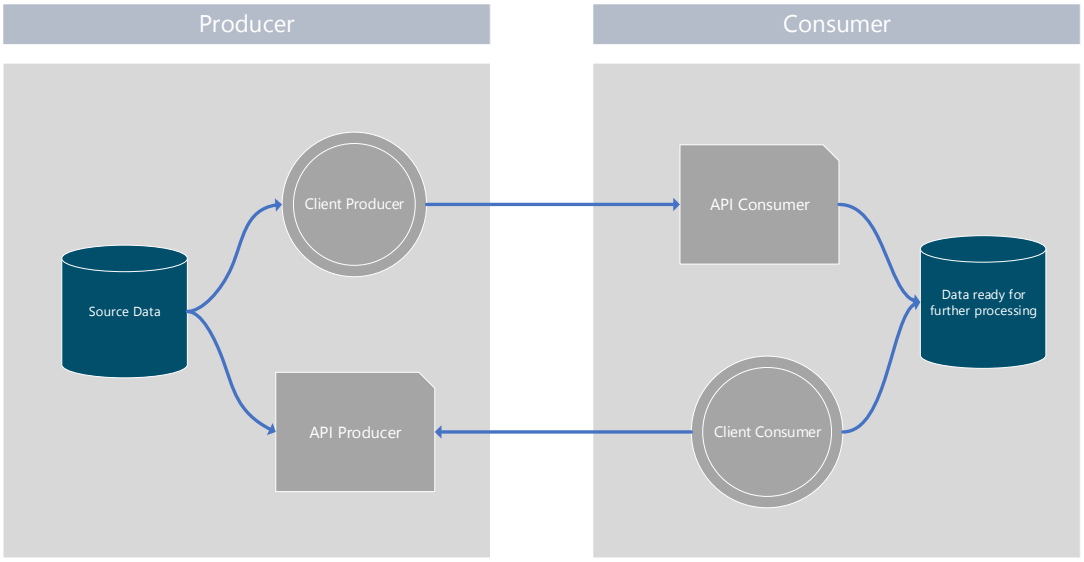


Figure 2 System overview

Each Consumer and Producer will have an Application Programming interface (API) that listens to and handles configuration and other messages, and a client that dispatches messages to its corresponding API. Together they form a DataExchange system.

The API will be REST-web services. The protocol used is JSON (ECMA-4-4) over HTTP/1.1. Vehicle data will be provided in EN-15430 format converted to JSON enhanced with meta info like modem serial number and record identifier.

The Consumers client will connect to the Producers API to manage configuration information and complete (missing) data.

The Producers API will provide an interface to manage configuration information and to get vehicle data.

The Producers client will Post (push) new vehicle data to the Consumers API.

The Consumers API will provide an interface to receive new vehicle data.

With this DataExchange system (from here on called DataExchange) it is possible to stream vehicle data to partner systems. It is a one-way data stream.

When configuring API and client systems to assume both the Consumer and the Producer roles it is possible to create a two-way system.

A Keep-Alive Message must be posted every minute to the Consumer. A Keep-Alive message is a message with SerialNumber Id 1 and does not contain EN-15430 data. Using this Keep-Alive Message, the Consumer can check if the post mechanism is running. If no Keep-Alive messages are received, the Consumer must switch to a get mechanism, and the Consumer must notify the support parties.

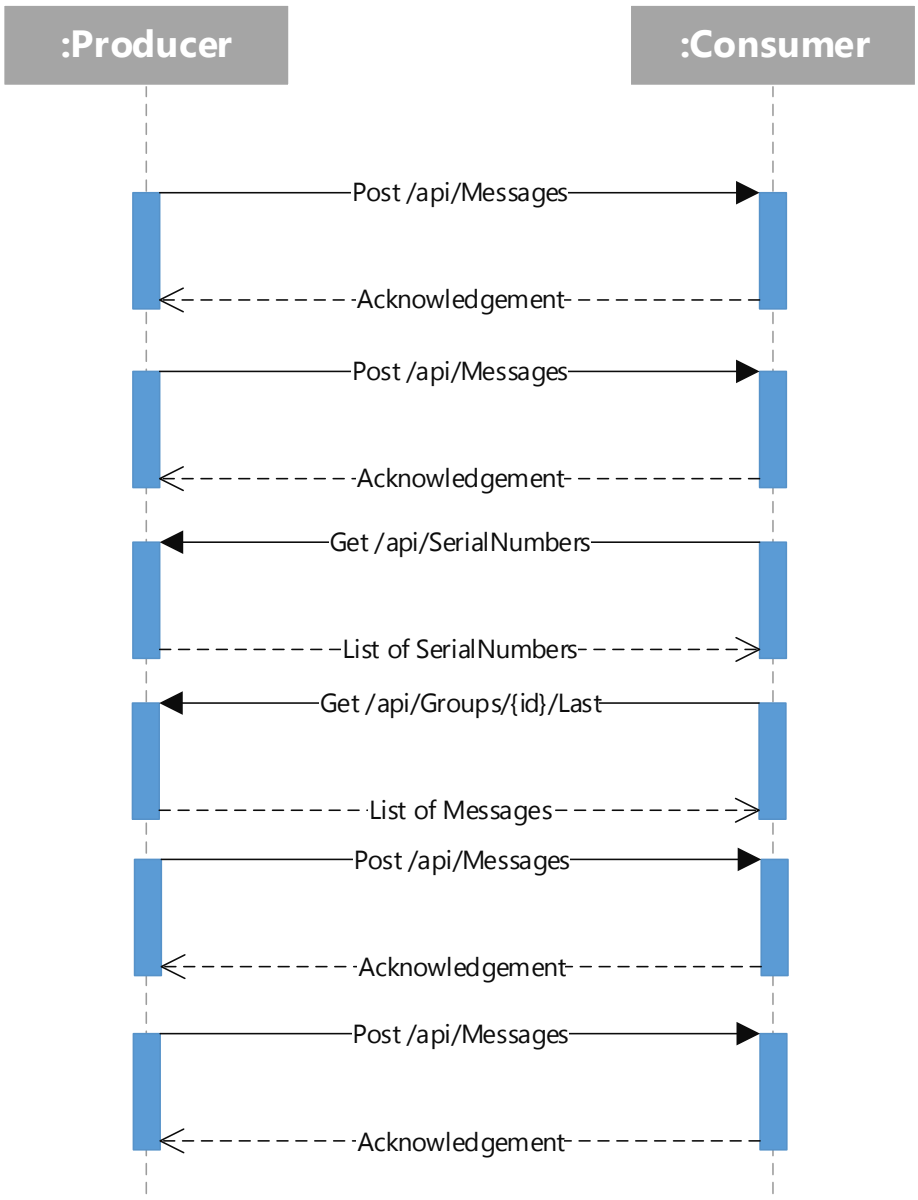
7.1 Flow

7.1.1 Normal operation flow

Normal operational flow is shown in the following sequence diagram.

Figure 3 Normal operation flow

New data will be posted by the Producers client at a maximum rate of once per second. When there is no new data there will be no posts. The Consumers API will check the Id for gaps in the data flow and other parameters of the data and store/process it. New



data will only be posted once. Every minute the Consumers client process request the last messages for all the serial numbers it is expecting data from and checks if all data is complete. At most once per 60 seconds, should be once every hour or even once a day, the Consumers client process requests if there are any new serial numbers available.

7.1.2 Missing messages detected flow

In case the Consumers API detects data is missing normal flow is interrupted.

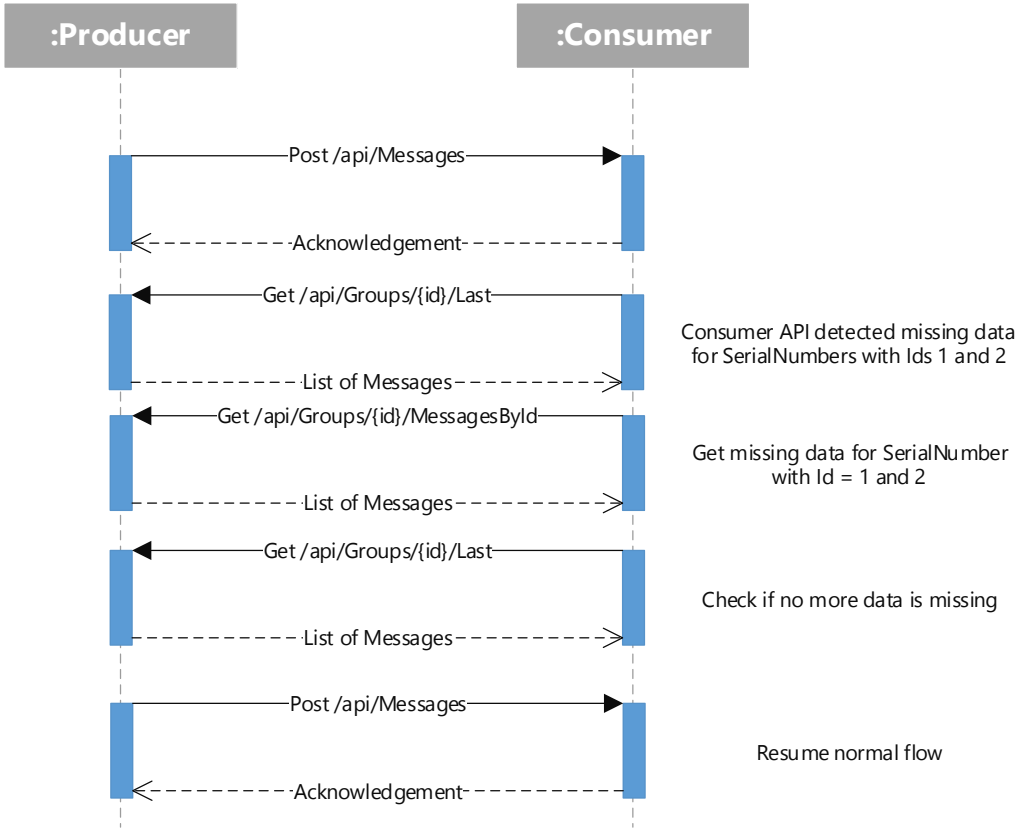
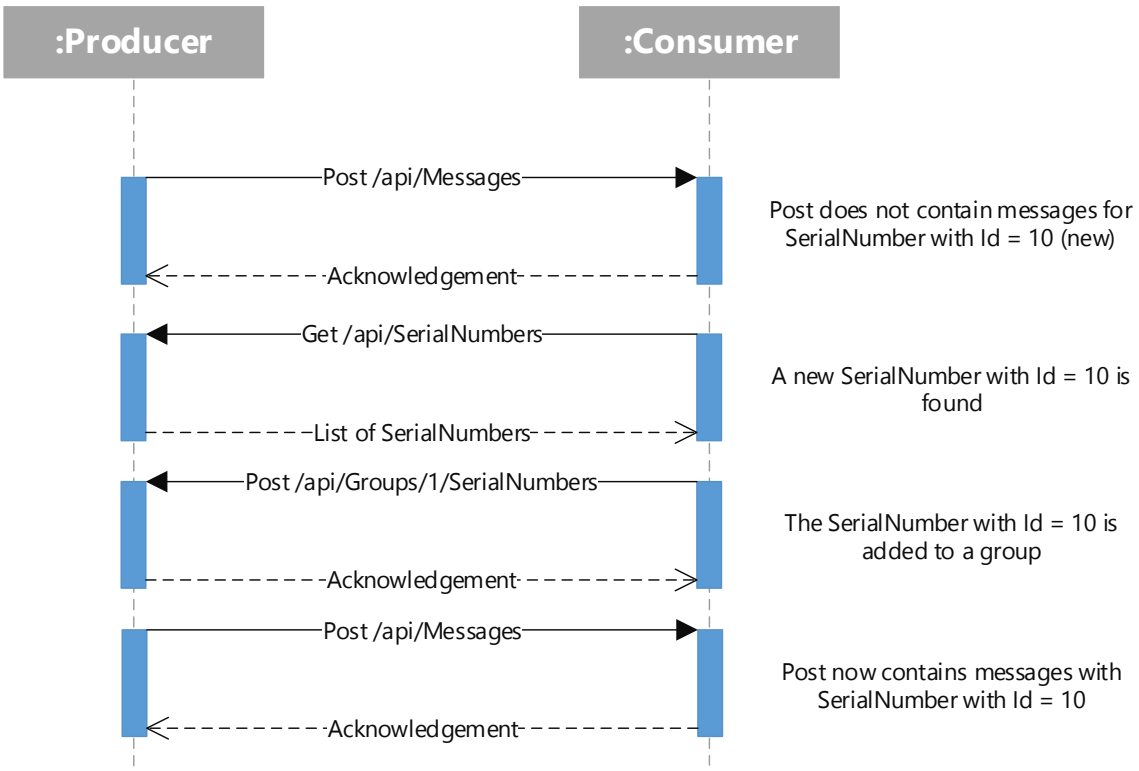


Figure 4 Missing messages detected flow

The Consumers API will ignore any messages posted by Producers client for the serial numbers where data is missing. Instead, it will for each serial number in sequence start getting the missing data, starting with the lowest index missing (oldest message). When all messages for a given serial number are complete, posts will be accepted again for that serial number.

7.1.3 *A new serial number is available flow*

In case the Consumers client detects that a new serial number is available for message retrieval, normal program flow is not interrupted. The following sequence diagram will



apply.

Figure 5 A new SerialNumber is available flow

When the Consumers client system detects a new serial number it will add that number to a default group with serial numbers. As a result of adding the serial number to a group the Producers client will start posting messages to the Consumers API. Posting messages for this new serial number will start with the first message that is presented to the Producers system after the number is added to the group. This to prevent that the system will be flooded with old data. When older data is needed, action is required by a person supporting the system.

8 Interface Design

DataExchange is a protocol that uses REST, JSON and HTTPS to be able to send machine data quick and reliable from parties collecting vehicle data to parties processing it. In essence, functionality of the interface can be divided into two parts. Functions about how and what vehicle data to send and functions sending the vehicle data. Vehicle data is sent in a JSONified version of EN-15430-1. The vehicle data records are enriched with the serial number of the board computer and two Id fields to guard data completeness.

8.1 Time information handling

In contrast to true EN-15430-1 data all date, time and timestamp fields in the described interface have to contain UTC. The timestamp shall be the true time of the creation of the message, for example the GPS timestamp in UTC but not the internal clock of the board computer. Local timestamps are avoided to be able to compare data generated in different time zones.

8.2 Message structure

Message are built according to the REST protocol definition. Messages must be lowercase. Capitals are not allowed. Message type and parameters are contained in the message URL. For example, to get the serial numbers of all the vehicles contained in the group with id 3 you would use: https://examplecompany.com/api/groups/3. Data is contained in the Body part of the HTTPS request. In case of the above example:

```
{
  "id": 0,
  "serialnumbers": [],
  "name": "Example Company"
}
```

Figure 6 Body part of a group{id} message

8.3 Acknowledgement

Each exchange message is responded to by the receiving system with a HTTP status code following HTTP protocol. Additional HTTP status codes are added to a message to provide extra information e.g. 404 stands for group not found when asking for a group with an unknown id. See RFC 2616 Chapter 10 for a complete overview of HTTP status codes.

8.4 Amendments to EN155430-1

Some small amendments to EN15430-1 have to be made. Often data coming in from vehicles use non en15430-1 protocols. This data has to be translated to the en15430-1 protocol resulting in rounding errors. And often fields are not available and have to be derived from other data. See appendix F.

8.5 Requests

The requests in the protocol are grouped into six categories: Groups. SerialNumbers, Messages, Logs, Time and Version. The API should provide a test interface, for example Swagger[1].

Figure 7 API Overview

8.5.1 Groups

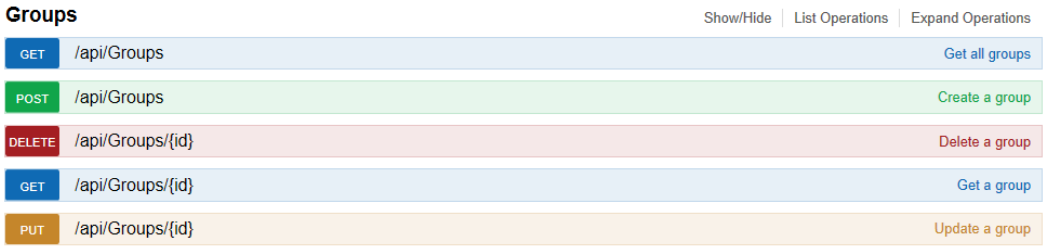


Figure 8 Group requests

[1] https://swagger.io/

A group contains serial numbers. Groups are used to enquire large numbers of vehicles in one API call to reduce the total number of API calls needed.

Body

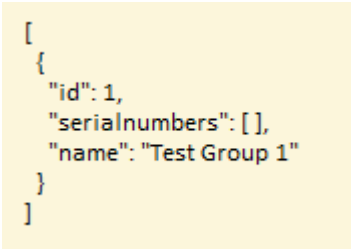


Figure 9 Group object body

Parameter	Explanation	Reference
Id	Unique identifier for group	
Serialnumbers	Array of all serial numbers in the group	See 6.4.2
name	Name of the group	

Table 1 Group object parameters

Available requests:

(Get : /api/groups)

Get a list of all available groups including the serial numbers those groups contain.

Extra Response Messages:

404 No groups found.

(POST : /api/groups)

Create a new group. A body containing the new group name and eventual serial numbers is added to the request. The Id provided in that body has to be set to 0. A new Id will be provided by the API.

Extra Response Messages:

400 Group id invalid.

(Delete : /api/groups/{id})

Delete the group with id: {id}. {id} shall be replaced with the id of the group that should be deleted.

Example: https://examplecompany.com/api/groups/3

Extra Response Messages:

204 Group is deleted.

(GET : /api/groups/{id})

Get the group with id: {id}. {id} shall be replaced with the Id of the group that is requested.

Extra Response Messages:

404 Group not found.

(Put : /api/groups/{id})

Change the name of the group with id: {id}. {id} shall be replaced with the id of the group of which the name has to be changed.

Extra Response Messages:

204 Group is updated.

400 Group is invalid.

404 Group not found.

8.5.2 Serial numbers (SerialNumbers)

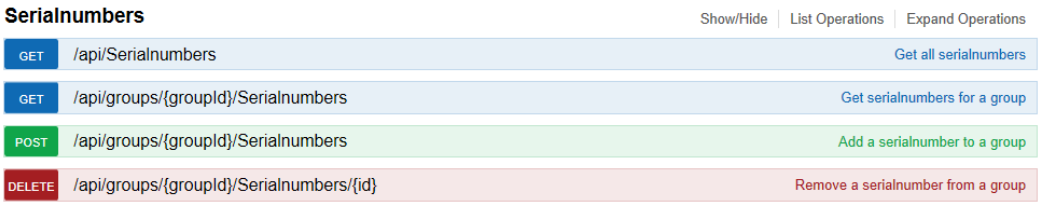
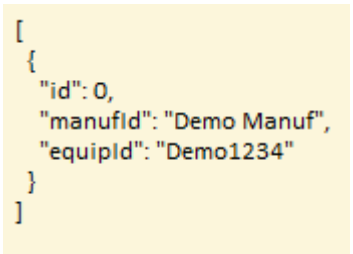


Figure 10 SerialNumer Requests

Each serial number contains an Id, a manufId and an equipId. The Id is an unique identifier used as a link to the datasource as identified by the manufId and equipId. The combination of manufId and equipId shall also be unique over a Producer/Consumer pair. The Id is used for all calls requesting data and as serial number identifier in groups in this protocol.



```
[
  {
    "id": 0,
    "manufId": "Demo Manuf",
    "equipId": "Demo1234"
  }
]
```

Body

Figure 11 SerialNumber object body

Parameter	Explanation	Reference
Id	Unique identifier for the Serial number	
manufId	Manufacturer identification.	EN 15430-1:2007+A1:2011 Table 6 No 5.
equipId	Manufacturers serial identification code of vehicle/equipment	EN 15430-1:2007+A1:2011 Table 6 No 6.

Table 2 SerialNumber object parameters

Available requests:

(Get : /api/serialnumbers)

Get a list of all available serial numbers. This list is provided for by the Producer and cannot be changed by the Consumer.

Extra Response Messages:

404 No serial numbers found.

(GET : /api/groups/{groupid}/serialnumbers)

Get a list of all available serial numbers contained in the group with id: {groupid}. {groupid} shall be replaced with the id of the group the serial numbers are requested for.

Example: https://examplecompany.com/api/groups/1/serialnumbers

Extra Response Messages:

404 Group not found.

(Post : /api/groups/{groupid}/serialnumbers)

Add a serial number to the group with Id: {groupid}. {groupid} shall be replaced with the id of the group the serial numbers should be added to. The serial number added has to be a member of the list provide by (Get : /api/serialnumbers)

Extra Response Messages:

404 Group or serial number not found.

(DELETE : /api/groups/{groupid}/serialnumbers/{id})

Delete the serial number with id {id} from the group with id: {groupid}. {groupid} shall be replaced with the Id of the group the serial numbers have to be deleted from. The serial number added has to be a member of the list provide by (Get : /api/serialnumbers)

Example: https://examplecompany.com/api/groups/1/serialnumbers/1

Extra Response Messages:

204 The serial number is removed from the group.

404 Group or serial number not found.

8.5.3 *Messages*

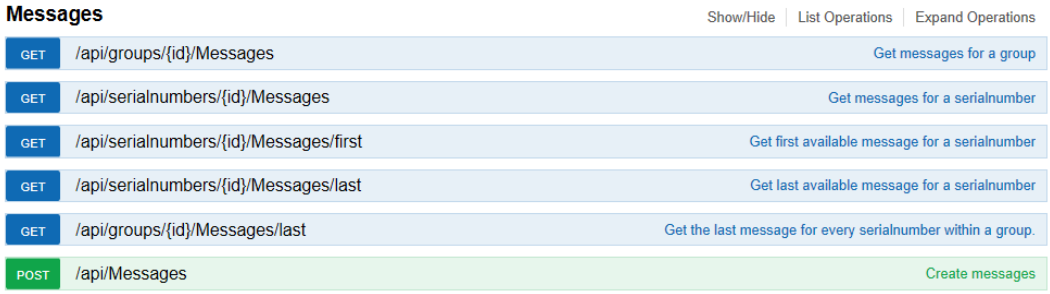


Figure 12 Messages Request Overview

A message contains

The message has

- SerialnumberId
- Id
- LastvalidId
- Header record
- Footer record
- Gps record
- Spreader record
- Snowplough Broom records

```
{
  "serialNumberId": 0,
  "id": 0,
  "lastValidId": 0,
  "header": {
    "version": "string",
    "driverId": "string",
    "driver2Id": "string",
    "routeId": "string",
    "id": 0,
```

Figure 13 Partial example of Message object body

The id is unique per serial number. The id is a sequence. Messages have to be sent in an ascending order. Gaps are not allowed. The lastvalidid is the last valid id of the previous message. In most case this property will be equal to the value of the id minus 1. Valid gaps can be defined by assigning a specific value to the lastvalidindex.

Records are records as defined in EN 15430-1 with two exceptions. The record string is converted to a JSON object and the fields id and bcId are added. Id is identical to the Id field of the Message. BcId is identical to the BC_ID field as described in CEN/TS 15430-2:2012 Table 1. It is a unique identifier string for the board computer that collects the data. Only one record of each type can be present in a message except for Snow plough Broom records which is an array. Often a vehicle has more than one plough and/or broom present.

In general, one message contains one EN 15430-1 record though this is optional.

Body


```
[
 {
  "serialNumberId": 1,
  "id": 327332,
  "lastValidId": 327331,
  "header": null,
  "gps": null,
  "spreader": null,
  "snowploughBrooms": [],
  "footer": null
 }
]
```


Figure 14 Message Body

Parameter	Explanation	Reference
serialNumberId	Id of the serial number this message belongs to.	
Id	Unique number in combination with SerialNumberId that identifies this message.	
LastValidId	Id of the message this message should follow up to. This Field makes it possible to detect if data is missing.	EN 15430-1:2007+A1:2011 Table 6 No 6.
header	EN 15430 header record	Appendix B.
gps	EN 15430 gps record	Appendix B.
spreader	EN 15430 spreader record	Appendix B.
snowploughbroom	Array of EN 15430 snowploughbroom records	Appendix B.
footer	En 15430 header record	Appendix B.

Table 3 Message object parameters

Available requests:

(Get : /api/groups/{id}/messages)

Returns all messages for the group with id: {id} over a time range. {id} shall be replaced with the id of the group for which the messages should be returned. The time range shall be provided through query parameters.

Query Parameter	Description	Data type
StartDateTime	Start of the time range (date and time) to filter the messages by.	Date-time (ISO 8601 format)
EndDateTime	End of the time range (date and time) to filter the messages by.	Date-time (ISO 8601 format)

Table 4 Query parameters /api/Groups/{Id}/Messages

Extra Response Messages:

400 Request is invalid.

404 No messages found.

(Get : /api/groups/{id}/messages)

Returns all messages for the group with id: {id} over an index range. {id} shall be replaced with the id of the group for which the messages should be returned. The index range shall be provided through query parameters. The index range refers to the id of the message.

Query Parameter	Description	Data type
startIndex	Start of the index range to filter the messages by.	64 bit integer
endIndex	End of the index to filter the messages by.	64 bit integer

Table 5 Parameters /api/Groups/{Id}/Messages

Example: https://examplecompany.com/api/groups/1/messages with startIndex = 1 and endIndex = 10

Will return all messages for all serial numbers which are member of group 1 that have an Id greater or equal to 1 and lower or equal than 10.

Extra Response Messages:

404 No messages found.

(Get : /api/serialnumbers/{id}/messages)

Returns all messages for the serial number with id: {id} over an index range. {id} shall be replaced with the id of the serial number for which the messages should be returned. The index range shall be provided through query parameters. The index range refers to the id of the message.

Query Parameter	Description	Data type
startIndex	Start of the index range to filter the messages by.	64 bit integer
endIndex	End of the index to filter the messages by.	64 bit integer

Table 6 Parameters /api/SerialNumbers/{Id}/Messages

Example: https://examplecompany.com/api/serialnumbers/1/messages with startIndex = 1 and endIndex = 10

Will return all messages for the serial number with (serial number) id = 1 that have an (message) id greater or equal to 1 and lower or equal than 10.

Extra Response Messages:

404 No messages found.

(Get : /api/serialnumbers/{id}/messages/first)

Returns the message for the serial number with id: {id} with the lowest (= the first) message id. {id} shall be replaced with the id of the serial number for which the message

should be returned. This request can be used to determine the oldest data that can be retrieved for a serial number by the Data Exchange mechanism.

Extra Response Messages:

404 No messages found.

(Get : /api/serialnumbers/{id}/messages/last)

Returns the message for the serial number with id: {id} with the highest (= the last) message id. {id} shall be replaced with the Id of the serial number for which the message should be returned. This request is used to determine the id of the latest created message and can be used to check if all data is received.

Extra Response Messages:

404 No messages found.

(Get : /api/groups/{id}/messages/last)

Returns for each serial number in the group with id: {id} the message with the highest (= the last) message id. {id} shall be replaced with the id of the group for which the message should be returned. This request is used to determine the id of the latest created messages in a group and can be used to check if all data is received.

Extra Response Messages:

404 No messages found.

(Get : /api/groups/{id}/messagesbyid)

Returns for each serialnumber requested a list of messages with ids between the requested range, including the start and end id. {id} shall be replaced with the id of the group for which the message should be returned.

Query Parameter	Description	Data type
SerialNumbers	A list of serialnumbers with a start and end id	A list of objects containing 64 bit integers

Table 7 Parameters /api/Groups/{Id}/MessagesById

Extra Response Messages:

404 No messages found.

(Post : /api/messages)

Create messages. This request is used to receive messages near real time.

8.5.4 Logs

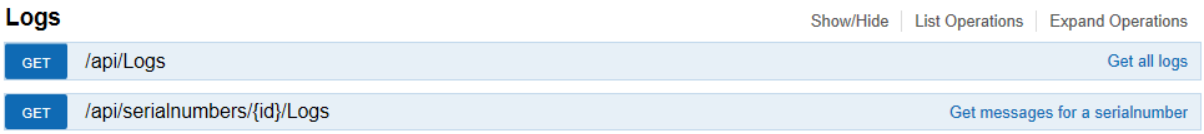


Figure 15 Log Request overview

Logs are added to provide some extra information when the API is failing. For example, a when a record cannot be sent because it contains invalid data.

Available requests:

(Get: /api/logs)

Get logs between a start and end datetime.

Query Parameter	Description	Data type
StartDateTime	Start of the time range (date and time) to filter the messages by.	Date-time (ISO 8601 format)
EndDateTime	End of the time range (date and time) to filter the messages by.	Date-time (ISO 8601 format)

Table 8 Parameters /api/Logs

(Get: /api/serialnumbers/{id}/logs)

Get logs for a specific serial number between a start and end datetime.

Query Parameter	Description	Data type
StartDateTime	Start of the time range (date and time) to filter the messages by.	Date-time (ISO 8601 format)
EndDateTime	End of the time range (date and time) to filter the messages by.	Date-time (ISO 8601 format)

Table 9 Paramers /api/SerialNumbers/{id}/Logs

8.5.5 Time

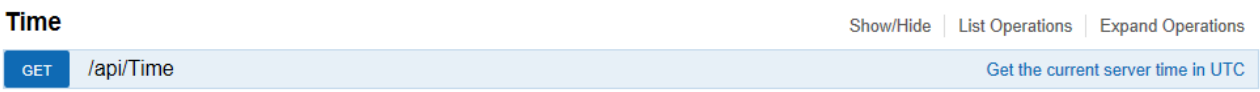


Figure 16 Time request overview

Time is provided to be able to synchronise Producer and Consumer systems in case one of them does not have access to a reliable time source. Also, it can be used to check if the corresponding server has the correct time.

Available request:

(Get: /api/time)

Get the current UTC datetime of the server in ISO 8601 format.

8.5.6 Version

(Get: /api/Version/API)

Get the current version of the API. This request returns a string with the version, e.g. “1.0.0.0”.

(Get: /api/Version/NEN)

Get the minimum and maximum versions of EN-15430-1 supported by the API. This request returns an object with the minimum and maximum version. Example :


```
{
    "minimum_version": "EN 154301: 2007 + A1: 2011",
    "maximum_version": "EN 154301 A1:2015"
}
```


(Get: /api/Version/Protocol)

Get the minimum and maximum versions of this protocol document support by the API. This request returns an object with the minimum and maximum version. Example:


```
{
    "minimum_version": "1.0.0.0",
    "maximum_version": "2.0.1.2"
}
```

9 Classification, designation and coding

9.1 Data integrity

Data exchange relies on REST and HTTP protocol to make sure data arrives unaltered by its clients. Messages are always sent complete or not at all. This does not prevent data from being incomplete as a result from for example system failure or network failure. In those cases, messages will probably be missing.

To check for and retrieving missing messages each message is supplied with two fields, Id and LastValidId. Id identifies the message itself uniquely, together with the Id of the serial number. LastValidId is the Id of the previous (valid) message for the same serial number. The Consumer compares LastValidId with the Id of the last received message for the same serial number. If the numbers do not match data is missing and action to retrieve the missing data should be taken. See 3.1.2 for a further description of this system.

9.2 Versioning

This document, protocol used, API, Client and all other components are versioned. Any change in document, protocol or code will always increase the version number used.

API request revealing the version numbers of the oldest and the newest versions of the protocol supported by the API. Consumer and Producer shall negotiate supported versions and use the highest version they both supports.

Appendix A

This appendix describes guidelines for the Winter and road service area maintenance equipment Data Exchange.

The guidelines below must always be agreed upon by both the Consumer and the Producer.

Performance

- The Producer may only send 5000 records per requests and the Consumer can only retrieve 5000 records per requests.
- Consumer should do no more than 1 request per second.
- The response times of the API should be less than 1 second.
- Messages should have a maximum delay of 10 seconds before posting to a Consumer.
- For getting data older than two years the response times do not apply.

Best Practices

- The data must be kept available for at least two years in a live database and up to five years in an archive.
- Each API must have a test interface, e.q. Swagger.
- If the Consumer API is not available, the Producer should send an automated notification to the correct support parties. The same applies the other way around.
- Empty fields should be omitted from the JSON.
- Messages including non-supported EN-15430-1 records should be ignored.
- Free definable EN-15430-1 records and variables must be added to appendix E.

Appendix B

Message body voorbeelden.

Header record


```
{
   "serialNumberId": 286730253,
   "id": 810499,
   "lastValidId": 810498,
   "header": {
     "version": "0",
     "driverId": null,
     "driver2Id": null,
     "routeId": null,
     "id": 810499,
     "bcId": "286730253",
     "manufId": "Nen15439DataExchange",
     "equipId": "S2B37278",
     "sysTime": "2040092",
     "sysDate": "0550434",
     "source": 5
   },
   "gps": null,
   "spreader": null,
   "snowploughBrooms": [],
   "footer": null
 }
```


Footer record


```
 {
   "serialNumberId": 286641116,
   "id": 459595,
   "lastValidId": 459594,
   "header": null,
   "gps": null,
   "spreader": null,
   "snowploughBrooms": [],
   "footer": {
     "version": "0",
     "driverId": "",
     "driver2Id": "",
     "routeId": "",
     "id": 459595,
     "bcId": "286641116",
     "manufId": "2090",
     "equipId": "S2B37835",
     "sysTime": "0437156",
```

```
    "sysDate": "0560234",
    "source": 5
   }
  }
```


GPS record


```
{
   "serialNumberId": 286641117,
   "id": 358324,
   "lastValidId": 358323,
   "header": null,
   "gps": {
    "geoLat": "5108.4615N",
    "geoLon": "00409.0602E",
    "geoAlt": 232,
    "geoSpd": 0,
    "geoCours": 44032,
    "geoTime": "0428236",
    "geoDate": "0560234",
    "geoSQ": 2,
    "geoSats": 12,
    "id": 358324,
    "bcId": "286641117",
    "manufId": "2090",
    "equipId": "S2B38709",
    "sysTime": "0428236",
    "sysDate": "0560234",
    "source": 5
   },
   "spreader": null,
   "snowploughBrooms": [],
   "footer": null
  }
```


Keep-Alive record


```
[
  {
   "serialNumberId": 1,
   "id": 1,
   "lastValidId": 0,
   "header": null,
   "gps": null,
   "spreader": null
   "snowploughBrooms": null,
   "footer": null
  }
]
```


Spreader record with 3 SnowploughBroom records

```
[
  {
    "serialNumberId": 302624161,
    "id": 850255,
    "lastValidId": 850254,
    "header": null,
    "gps": null,
    "spreader": {
      "deviceEntity": 255,
      "sprMode": 0,
      "sprSimSpd": 0,
      "sprWidthSet": "11.5m",
      "sprSymSet": "10m|1.5m",
      "sprDosSet": "0|0|0",
      "sprWidth": 92,
      "sprWiLe": 80,
      "sprDosRes1": 0,
      "sprDosRes2": 0,
      "sprDosBr_g": 0,
      "sprDosBr_ml": 0,
      "sprBrinePerc": 0,
      "sprMaxOn": "00",
      "sprCntRes1": 21622,
      "sprCntRes2": 0,
      "sprCntBrineL": 25398,
      "sprCntBrineKg": 30478,
      "sprCntLen": 14060000,
      "sprCntHrs": 5,
      "drivenLen": 5738,
      "runHrs": 10,
      "vehSpd": 15360,
      "sprErr": "01",
      "sprErrCode": "2",
      "sprErrRpt": "",
      "geoLat": "5537.4052N",
      "geoLon": "01205.3713E",
      "geoAlt": 65535,
      "geoSQ": 2,
      "sprMatRes1": "3",
      "sprMatRes2": "1",
      "sprBrineTyp": "129",
      "sprThGr": "0",
      "sprEngMode": 255,
      "sprEngHrs": 4294967295,
      "airHum": 255,
      "airTemp": 255,
      "roadTemp": 255,
      "roadFrict": 65535,
      "beaconOn": "11",
      "sprOilT": 255,
      "sprOilP": 255,
      "sprOilQ": 4294967295,
```

```
  "id": 850255,
  "bcId": "302624161",
  "manufId": "2090",
  "equipId": "S3B10829",
  "sysTime": "1242212",
  "sysDate": "1020234",
  "source": 5
},
"snowploughBrooms": [
  {
    "recordCode": 6,
    "deviceEntity": 1,
    "pbPresent": "01",
    "pbPos": 1,
    "pbMode": 0,
    "pbCntLen": 0,
    "pbCntHrs": 0,
    "drivenLen": 5738,
    "runHrs": 10,
    "vehSpd": 15360,
    "geoLat": "5537.4052N",
    "geoLon": "01205.3713E",
    "geoSQ": 2,
    "pbRelease": "ÿ",
    "pbRelPerc": 255,
    "pbInfo": "ÿ",
    "bmRotSpd": 255,
    "pbOilT": 255,
    "pbOilP": 255,
    "pbOilQ": 4278190080,
    "brmRotSpd": 255,
    "id": 850255,
    "bcId": "302624161",
    "manufId": "2090",
    "equipId": "S3B10829",
    "sysTime": "1242212",
    "sysDate": "1020234",
    "source": 5
  },
  {
    "recordCode": 6,
    "deviceEntity": 2,
    "pbPresent": "01",
    "pbPos": 2,
    "pbMode": 0,
    "pbCntLen": 0,
    "pbCntHrs": 0,
    "drivenLen": 5738,
    "runHrs": 10,
    "vehSpd": 15360,
    "geoLat": "5537.4052N",
    "geoLon": "01205.3713E",
```

```
      "geoSQ": 2,
      "pbRelease": "ÿ",
      "pbRelPerc": 255,
      "pbInfo": "ÿ",
      "bmRotSpd": 255,
      "pbOilT": 255,
      "pbOilP": 255,
      "pbOilQ": 4278190080,
      "brmRotSpd": 255,
      "id": 850255,
      "bcId": "302624161",
      "manufId": "2090",
      "equipId": "S3B10829",
      "sysTime": "1242212",
      "sysDate": "1020234",
      "source": 5
    },
    {
      "recordCode": 6,
      "deviceEntity": 3,
      "pbPresent": "01",
      "pbPos": 3,
      "pbMode": 0,
      "pbCntLen": 0,
      "pbCntHrs": 0,
      "drivenLen": 5738,
      "runHrs": 10,
      "vehSpd": 15360,
      "geoLat": "5537.4052N",
      "geoLon": "01205.3713E",
      "geoSQ": 2,
      "pbRelease": "ÿ",
      "pbRelPerc": 255,
      "pbInfo": "ÿ",
      "bmRotSpd": 255,
      "pbOilT": 255,
      "pbOilP": 255,
      "pbOilQ": 4278190080,
      "brmRotSpd": 255,
      "id": 850255,
      "bcId": "302624161",
      "manufId": "2090",
      "equipId": "S3B10829",
      "sysTime": "1242212",
      "sysDate": "1020234",
      "source": 5
    }
  ],
  "footer": null
 }
]
```

Appendix C Requirements

DAR01	Authentication for all API calls is required (excluding DAR20 and DAR21)
DAR02	The API has to be implemented as a REST service

Groups

DAR03	The interface contains a request to get all groups
DAR04	The interface contains a request to create a new group
DAR05	The interface contains a request to delete a group
DAR06	The interface contains a request to retrieve a single group
DAR07	The interface contains a request to update a group

Logs

DAR08	The interface contains a request to retrieve logs between a start and end timestamp
DAR09	The interface contains a request to retrieve logs between a start and end timestamp for a specific serialnumber

Messages

DAR10	The interface contains a request to retrieve the last message for every serialnumber in a group
DAR11	The interface contains a request to retrieve all messages from a group between a start and end timestamp
DAR12	The interface contains a request to retrieve all messages for a serialnumber between a start and end index
DAR13	The interface contains a request to retrieve the first message of a serialnumber
DAR14	The interface contains a request to retrieve the last message of a serialnumber
DAR15	The interface contains a request to insert multiple messages

SerialNumbers

DAR16	The interface contains a request to retrieve all available serialnumbers
DAR17	The interface contains a request to retrieve the serialnumber in a group
DAR18	The interface contains a request add a serialnumber to a group

DAR19	The interface contains a request to remove a serialnumber from a group

Time

DAR20	The interface contains a request to get the current server timestamp

Version

DAR21	The interface contains a request to get the current versions used by the API

General requirements

DAR22	Requirements DAR03 to DAR19 shall only be performed for the associated authenticated account
DAR23	All received Messages have to be validated according to NEN-EN 15430-1:2015
DAR24	Only Messages from SerialNumbers added to a group shall be send

Background processes

DAR25	A maximum of 1 request per second is allowed
DAR26	A process shall queue new data and sent this data to every consumer
DAR27	If data could not be sent for more than 10 minutes, a notification shall be sent to both producer and consumer
DAR28	Data shall always be delivered in the correct order (ID of the messages)
DAR29	If received data is missing messages, this data has to be retrieved from the producer
DAR30	Periodically (with a maximum frequency of once per minute), a data check shall be performed to ensure the consumer received the latest data of the producer. If not, the consumer has to pull this data
DAR31	If the consumer is not able to receive data from the API of the producer for more than 10 minutes, a notification shall be sent to both producer and consumer
DAR32	If the average response time is more than 10 seconds for at least 5 minutes, a notification shall be sent to both producer and consumer
DAR33	All data fields available from the machine and described in the EN-15430 protocol have to be available in the API

Appendix D Where to get the documentation

CSN EN 15430-1:2007 + A1:2011 can be bought at:

https://www.en-standard.eu/csn-en-15430-1-winter-and-road-service-area-maintenance-equipments-data-acquisition-and-transmission-part-1-in-vehicle-data-acquisition/

CSN EN 15430-1 A1:2015 can be bought at:

https://www.en-standard.eu/csn-en-15430-1-winter-and-road-service-area-maintenance-equipments-data-acquisition-and-transmission-part-1-in-vehicle-data-acquisition/

(UNE) CEN/TS 15430-2:2012 can be bought at:

https://www.en-standard.eu/une-cen-ts-15430-2-2012-winter-and-road-service-area-maintenance-equipment-data-acquisition-and-transmission-part-2-protocol-for-data-transfer-between-information-supplier-and-client-application-server-endorsed-by-asociacion-espa-ola-de-normalizacion-in-s/

FIELDING, Roy, et al. RFC 2616: Hypertext transfer protocol–HTTP/1.1. 1999

https://www.w3.org/Protocols/rfc2616/rfc2616.html

Appendix E Adaptation and filling of free Data records/variables

The tables in this Appendix describe free records as described in EN 15430-1.

Spreader automation record

The spreader automation record is a reservation for future use. It is not obligated for version 1.1 and earlier of this document.

Table 1 — Spreader automation variables

Name	Description	BASIC data format or SLOT
SaMode	Automatic spreading is active (0=no, 1=yes).	UNSIGNED CHAR
SaManOverride	Are the settings proscribed by the automatic spreading system manually overridden by the chauffeur. (0= no, 1=yes)	BOOLEAN

Table 2 — Spreader automation data

Code	Name	Mandatory/Optional	Trigger	Remark
10008	Spreader automation data	Optional	Start, End, Time, Field	This record is generated if there is a spreader/sprayer on the vehicle with such data generation facilities.
Field		**Mandatory/Optional**	**Trigger**	**Remark**
ManufID		Mandatory		Manufacturer identification of the system which generates this record.
EquipID		Mandatory		Spreader identification

DeviceEntity	Optional		If more than one implement is equipped to the vehicle, this number allows to differentiate between the individual implements.
Source	Mandatory		
SysTime	Mandatory		
SysDate	Mandatory		
SaMode	Mandatory	Any change	
SaManOverride	Mandatory	Any change	

Appendix F Calculation of missing fields and rounding of data.

When porting data from protocols, f.e. DAU, to EN15430-1 often fields or field values are missing. In those cases, these fields will have to be calculated from other fields when possible.

Also, field values often will have to be rounded resulting in small discrepancies. This appendix describes how to handle those fields.

SprMode

Table 5.5.8 **CSN EN 15430-1 A1:2015**

The SprMode value 'Spraying' is missing in many spreader protocols. It should be calculated using the mode field from those protocols (value is 'active' or 'spreading') together with the fields SprDosRes1, SprDosRes2, SprDosBr_g and SprBrinePerc (or equivalent fields from those other protocols).

When SprMode = spraying, Fluid (Brine) is spayed using a spray bar. The amount of fluid sprayed can be calculated by.

br = SprDosBr_g

s = SprDosRes1 + SprDosRes2

x = SprBrinePerc

Sprayed dosage = br – s * (x / (1 - x)).

When the 'Sprayed dosage' is > 0 and the spreader mode is 'active' or 'spreading' the new sprMode will be 'Spraying' when s = 0 and 'Spreading and Spraying' when s > 0.

Rounding problems make the above calculation tricky. It is possible that the 'Sprayed dosage' is > 0 when is should not be. To get around this problem a tolerance in the calculation is used. When the 'Sprayed dosage' is less than 1 % of the total dosage (s + br) it is assumed that the vehicle is not spraying and the resulting SprMode will be 'Spreading'.

Appendix G Material setting and names.

The Spreader record fields SprMatRes1, SprMatRes2, SprBrineTyp are optional in EN 15430-1. This is inconvenient for automated report creation by software. Therefore, in this document their status is changed to mandatory.

The similar problem arises with the definition of the material setting. Each manufacturer can use its own specific table of codes. This meaning the reporting program needs to know all those tables and will have to know how to link each of the materials in those tables with the materials to report.

To circumvent this, only materials in this appendix provided table are allowed. It is obligated to use the material code of the material provided in table 1 of this appendix. Other names/codes are not allowed unless agreed upon by both producer and consumer(s) of the WARSAME system.

The material codes that are of material type Dry are intended to use with SprMatRes1 and SprMatRes2. The material code of type Wet are to be used with SprBrineTyp.

New materials are to be added to the end of the table using a unique name and code.

Table 1 — Material codes

Material code	Material name	Relative density	Type of material
1	NaCl fine	1.2	Dry
2	NaCl medium	1.25	Dry
3	NaCl coarse	1.32	Dry
4	CaCl2 fine	1.3	Dry
5	CaCl2 medium	1.3	Dry
6	CaCl2 coarse	0.95	Dry
7	Urea	0.88	Dry
8	Safeway SD	0.69	Dry
9	DiMix	1	Dry

Material code	Material name	Relative density	Type of material
10	Clearway 25	1.28	Dry
12	Clearway 25	1.28	Dry
13	Clearway 25	1.28	Dry
14	NKKM	0.88	Wet
67	Grit	2	Dry
68	Medium sand < 0,5mm	1.6	Dry
69	Coarse sand < 1mm	1.6	Dry
70	Very coarse sand < 2mm	1.6	Dry
129	NaCl	1.16	Wet
130	CaCl2	1.16	Wet
131	MgCl2	1.16	Wet
132	Clearway 1	1.28	Wet
133	Safeway KA	1	Wet
134	Frigantine	1.27	Wet
135	Urea	1	Wet
136	Eco 6	1	Wet
137	K.K.O.	1	Wet
138	Potassium Acetate	1.28	Wet
139	Aviform	1	Wet
140	Glycol Alcohol	1	Wet
141	Potassium Acetate	1	Wet
142	Potassium Formate	1.33	Wet
143	Potassium Formate	1.35	Wet
144	Potassium Formate	1.34	Wet
146	Nordway	1.25	Wet
147	Potassium Formate	1.29	Wet
150	Potassium Formate(1P33,1P35)	1.34	Wet
151	Nordway	1.25	Wet
152	Water	1	Wet

Material code	Material name	Relative density	Type of material
153	Nordway Granulate SG	0.95	Dry
154	Nordway KF SG	1.33	Dry
155	Kemka Ice-Breaker	0.95	Wet
156	Nordway Granulat 1	1.2	Dry
157	Potassium Formate	1.32	Wet
255	Isomex	1.29	Wet
256	Glycomex	1.1	Dry
157	Sodium Acetate Granulate	1	Dry