



# ACS

## Ubisense ACS Protocol

Version 1.7

Part Number: ACS\_UBI\_PRO\_1.7\_EN

Copyright © 2023, Ubisense Limited 2014 - 2023. All Rights Reserved. You may not reproduce this document in whole or in part without permission in writing from Ubisense at the following address:

Ubisense Limited  
St Andrew's House  
St Andrew's Road  
Cambridge CB4 1DL  
United Kingdom

Tel: +44 (0)1223 535170

WWW: <https://www.ubisense.com>

All contents of this document are subject to change without notice and do not represent a commitment on the part of Ubisense. Reasonable effort is made to ensure the accuracy of the information contained in the document. However, due to on-going product improvements and revisions, Ubisense and its subsidiaries do not warrant the accuracy of this information and cannot accept responsibility for errors or omissions that may be contained in this document.

Information in this document is provided in connection with Ubisense products. No license, express or implied to any intellectual property rights is granted by this document.

Ubisense encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.

UBISENSE®, the Ubisense motif, SmartSpace® and AngleID® are registered trademarks of Ubisense Ltd. DIMENSION4™ and UB-Tag™ are trademarks of Ubisense Ltd.

Windows® is a registered trademark of Microsoft Corporation in the United States and/or other countries. The other names of actual companies and products mentioned herein are the trademarks of their respective owners.

# Contents

---

|   |          |
|---|----------|
| <b>Introduction</b> .....                       | <b>1</b> |
| <b>General message structure</b> .....          | <b>2</b> |
| <b>Client&lt;-&gt;Server Architecture</b> ..... | <b>4</b> |
| <b>Telegram Type</b> .....                      | <b>5</b> |
| List of Telegram Types .....                    | 5        |
| Telegram Types in Detail .....                  | 6        |
| ALIVE .....                                     | 6        |
| RESPONSE .....                                  | 7        |
| PRODTAG .....                                   | 8        |
| PRODTAG, version 01 .....                       | 9        |
| PRODTAG, version 02 .....                       | 11       |
| PRODTAG, version 03 .....                       | 13       |
| SYNCPROD .....                                  | 15       |
| SYNCPROD, version 01 .....                      | 16       |
| SYNCPROD, version 02 .....                      | 17       |
| SYNCPROD, version 03 .....                      | 18       |
| PRODDATA .....                                  | 19       |
| PRODDTRQ .....                                  | 21       |
| PRODAT .....                                    | 23       |
| Externally-driven Assembly Lines .....          | 25       |
| Trigger point telegram example .....            | 26       |
| ISINZ .....                                     | 27       |
| ISINZ Version 01 .....                          | 28       |
| ISINZ Version 02 .....                          | 29       |
| ISOUTZ .....                                    | 30       |
| ISOUTZ Version 01 .....                         | 31       |
| ISOUTZ Version 02 .....                         | 32       |
| ZONESTAT .....                                  | 33       |
| ZONESTAT Version 01 .....                       | 34       |
| ZONESTAT Version 02 .....                       | 35       |

# Contents

---

|                                     |    |
|-------------------------------------|----|
| ZONESTAT Version 03 .....           | 36 |
| LINESTAT .....                      | 37 |
| ISENABLE .....                      | 41 |
| SUBSCR .....                        | 42 |
| UNSUBSCR .....                      | 46 |
| EXCEPT .....                        | 48 |
| Example Messages .....              | 49 |
| EXCEPTRQ .....                      | 51 |
| EXCEPTRS .....                      | 54 |
| TAGPOS .....                        | 55 |
| TAGPOS Version 01 .....             | 55 |
| TAGPOS Version 02 .....             | 57 |
| TAGPOS Version 03 .....             | 59 |
| TAGPOS Version 04 .....             | 60 |
| TAGSTAT .....                       | 62 |
| TAGSTAT Version 01 .....            | 63 |
| TAGSTAT Version 02 .....            | 64 |
| TAGSTRQ .....                       | 65 |
| TAGSTRQ Version 01 .....            | 65 |
| TAGSTRQ Version 02 .....            | 66 |
| APPCONF .....                       | 67 |
| Configurations handled by ACS ..... | 68 |

# Introduction

---

This document describes the type of communication between the Ubisense Assembly Control System (ACS) and other systems (Server MES, PLC, referred to as External Systems).

Communication occurs through transmission of messages over TCP/IP. The communication interface is described in the following sections.

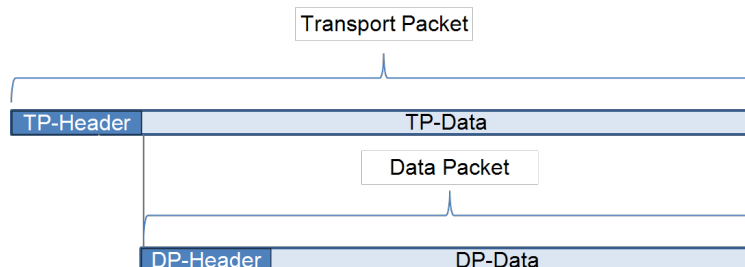


The document describes the current version of the protocol and is subject to change.

## General message structure

---

All messages transmitted via TCP/IP are constructed according to the scheme described below:



### Description

- All messages are transmitted in a transport packet (TP)
- The Transport Packet consists of a TP-Header and a TP-Data part
- The TP-Header always starts with the two characters "TH", followed by a two-digit Version Number
- The version number in the TP-Header determines what additional data is stored in the TP-Header. It also determines the length of TP-Data, and where TP-Data starts.
- The TP-Data are transmitted in a data packet (DP)
- The data packet consists of a DP-Header and a DP-Data part
- The DP-Header begins with the Telegram Type (8 characters) followed by a two-digit Version Number
- The Telegram Type and the Version Number in the DP-Header determine how TP-Data is processed / performed.

**Definition**

TP-Header Version 01

| Field                | Length | Type<br>(ascii / integer / float /<br>hex) | Alignment | Padding<br>characters | Example  |
|----------------------|--------|--|-----------|-----------------------|----------|
| Transport<br>Header  | 2      | ascii                                      | fixed     | none                  | TH       |
| TP-Version<br>Number | 2      | integer                                    | right     | '0'                   | 01       |
| TP length            | 8      | integer                                    | right     | '0'                   | 00000082 |

DP-Header Version 01

| Field                | Length | Type<br>(ascii / integer / float /<br>hex) | Alignment | Padding<br>characters | Example  |
|----------------------|--------|--|-----------|-----------------------|----------|
| Telegram Type        | 8      | ascii                                      | right     | '*'                   | ***ISINZ |
| DP-Version<br>Number | 2      | integer                                    | right     | '0'                   | 01       |

**Basics**

- Data telegrams require an acknowledgment telegram, if the data telegram is not a kind of request which results in a response anyway. If a response data telegram is expected, no separate acknowledgment is required
- Acknowledgments have their own message types defined
- If you experience data transmission errors, for example if telegrams are incorrect or incomplete, the connection is broken and restarted
- Timeout period for reception: 500 ms (configurable)
- Time for a connection: 500 ms (configurable)

The structure of the appended data is described in [Telegram Type](#).

## Client<->Server Architecture

---

ACS can be configured to act in one of two ways:

- As a TCP server, i.e. it provides a port for setting up a connection and waits for a connection request
- As a TCP client, i.e. it provides a connection request to a configured server (IP: Port)

As such ACS can be configured according to the TCP role that the external system (server MES or PLC) takes.



# Telegram Type

## List of Telegram Types

| Telegram Type | Description   |
|---------------|---|
| ALIVE         | Keep Alive telegram   |
| RESPONSE      | Acknowledgment  |
| PRODTAG       | Information about the association or disassociation of a product to a Ubisense tag                      |
| SYNCPROD      | Synchronization of product tag associations   |
| PRODDATA      | Information about product attributes  |
| PRODDTRQ      | Request to send additional product attributes   |
| PRODAT        | Information about products being at a location  |
| ISINZ         | Object in zone  |
| ISOUTZ        | Object left zone  |
| ZONESTAT      | List of objects currently contained in a zone / list of zones in which an object is currently contained |
| LINESTAT      | Status information of an assembly line, including the sequence of products and their positions          |
| ISENABLE      | Object enabled/disabled   |
| SUBSCR        | Subscribe for receiving event based information   |
| UNSUBSCR      | Remove formerly made subscriptions  |
| EXCEPT        | ACS information and warning messages  |
| EXCEPTRQ      | Request specific information and warning messages   |
| EXCEPTRS      | Response to request for information and warning messages  |
| TAGPOS        | Tag position messages   |
| TAGSTAT       | Tag battery status message  |
| TAGSTRQ       | Tag battery status request  |
| APPCONF       | Application configuration request   |

## Telegram Types in Detail

### ALIVE

#### Keep Alive telegram

This telegram is sent every 15 seconds if there has been no other telegram traffic. The 15 second timeout is a default value. Other values can be configured.

If the recipient does not reply to the telegram in less than 15 seconds, the connection is interrupted and must be restarted.

Sender: TCP client

Recipient: TCP server

Acknowledgment: yes

| Description                  | Type    | Length (bytes) | Alignment | Padding characters | Example  |
|------------------------------|---------|----------------|-----------|--------------------|----------|
| <b>Transport Packet (TP)</b> |         |                |           |                    |          |
| Transport Header             | ascii   | 2              | fixed     | none               | TH       |
| TP-Version Number            | integer | 2              | right     | '0'                | 01       |
| TP length                    | integer | 8              | right     | '0'                | 0000022  |
| <b>Data Packet (DP)</b>      |         |                |           |                    |          |
| Telegram Type                | ascii   | 8              | right     | '*'                | ***ALIVE |
| DP-Version Number            | integer | 2              | right     | '0'                | 01       |

#### Example

```
TH0100000022***ALIVE01
```

## RESPONSE

### Acknowledgment telegram

This telegram is sent in response to telegram types requiring acknowledgment.

Sender: TCP server or client

Recipient: TCP server or client

Acknowledgment: no

| Description                  | Type    | Length (bytes) | Alignment | Padding characters | Example                                       |
|------------------------------|---------|----------------|-----------|--------------------|---|
| <b>Transport Packet (TP)</b> |         |                |           |                    |   |
| Transport Header             | ascii   | 2              | fixed     | none               | TH  |
| TP-Version Number            | integer | 2              | right     | '0'                | 01  |
| TP length                    | integer | 8              | right     | '0'                | 00000028                                      |
| <b>Data Packet (DP)</b>      |         |                |           |                    |   |
| Telegram Type                | ascii   | 8              | right     | '*'                | RESPONSE                                      |
| DP-Version Number            | integer | 2              | right     | '0'                | 01  |
| Response Type                | integer | 2              | right     | '0'                | 00 = Not OK<br>01 = OK                        |
| Error code                   | integer | 4              | right     | '0'                | 0000 = No Error<br>0001 = Non Specified Error |

### Example

```
TH0100000028RESPONSE01010000
TH0100000028RESPONSE01000001
```

## PRODTAG

Information about the association or disassociation of a product to a Ubisense tag and the current state of the association process.

See version 01 below for a variant using a 4-byte tag ID in octet format, version 02 for a variant using a hexadecimal tag ID representation for a 4-byte tag ID, and version 03 for a variant which is capable of transferring an 8-byte tag ID in hexadecimal representation.

If ACS is sending PRODTAG telegrams, the DP-Version is automatically selected, dependent on the following factors:

- ACS Property 'ACS Protocol Service – UseHexTagID': If set to a value of '0' then DP-Version 01 is always used in PRODTAG telegrams.
- If a Ubisense Series 7000 location system is in use (both pure and in S7K-D4 co-existence) and if the 'ACS Protocol Service – UseHexTagID' property is set to '1', then DP-Version 02 is used in PRODTAG telegrams.
- If a Ubisense D4 location system is used (only pure, no S7K-D4 co-existence) and if the 'ACS Protocol Service – UseHexTagID' property is set to '1', then DP-Version 03 is used in PRODTAG telegrams.

Sender: ACS

Recipient: Control system (Server MES or PLC...)

Acknowledgment: yes

## PRODTAG, version 01

| Description                       | Type    | Length (bytes) | Alignment | Padding characters | Example                                       |
|-----------------------------------|---------|----------------|-----------|--------------------|---|
| <b>Transport Packet (TP)</b>      |         |                |           |                    |   |
| Transport Header                  | ascii   | 2              | fixed     | none               | TH  |
| TP-Version Number                 | integer | 2              | right     | '0'                | 01  |
| TP length                         | integer | 8              | right     | '0'                | 00000167                                      |
| <b>Data Packet (DP)</b>           |         |                |           |                    |   |
| Telegram Type                     | ascii   | 8              | right     | '*'                | *PRODTAG                                      |
| DP-Version Number                 | integer | 2              | right     | '0'                | 01  |
| Tag Circuit                       | ascii   | 32             | right     | '*'                | Name of the tag circuit                       |
| Association / Disassociation Zone | ascii   | 32             | right     | '*'                | Name of the association / disassociation zone |
| Product type information          | ascii   | 32             | right     | '*'                | *****MyProductType                            |

## Telegram Type

| Description        | Type    | Length (bytes) | Alignment | Padding characters | Example  |
|--------------------|---------|----------------|-----------|--------------------|--|
| Product name       | ascii   | 32             | right     | '*'                | *****MyProduct   |
| Tag-ID             | ascii   | 15             | right     | '*'                | 100-000-032-117 or 000-000-000-000   |
| Error status       | integer | 1              | right     | '0'                | 0 = No error<br>1 = No tag in association zone<br>2 = More than one tag in association zone<br>3 = No product information available<br>4 = Product information ambiguous<br>5 = Tag battery not OK<br>6 = Tag was recently already associated<br>7 = Product information is not reliable |
| Association status | integer | 1              | fixed     | none               | 0 = disassociated<br>1 = association in progress<br>2 = associated<br>3 = association canceled   |

### Association Example

```
TH0100000167*PRODTAG01*****MainLine*****4711
*****MyProductType*****MyProduct100-000-032-11702
```

### Disassociation Example

```
TH0100000167*PRODTAG01*****MainLine*****4712
*****MyProductType*****MyProduct000-000-000-01000
```

**PRODTAG, version 02**

This version uses a hexadecimal representation of a 4-byte tag ID (for a version 2 / S7K tag).

| Description                       | Type    | Length (bytes) | Alignment | Padding characters | Example                                       |
|-----------------------------------|---------|----------------|-----------|--------------------|---|
| <b>Transport Packet (TP)</b>      |         |                |           |                    |   |
| Transport Header                  | ascii   | 2              | fixed     | none               | TH  |
| TP-Version Number                 | integer | 2              | right     | '0'                | 01  |
| TP length                         | integer | 8              | right     | '0'                | 00000160                                      |
| <b>Data Packet (DP)</b>           |         |                |           |                    |   |
| Telegram Type                     | ascii   | 8              | right     | '*'                | *PRODTAG                                      |
| DP-Version Number                 | integer | 2              | right     | '0'                | 02  |
| Tag Circuit                       | ascii   | 32             | right     | '*'                | Name of the tag circuit                       |
| Association / Disassociation Zone | ascii   | 32             | right     | '*'                | Name of the association / disassociation zone |
| Product type information          | ascii   | 32             | right     | '*'                | *****MyProductType                            |

## Telegram Type

| Description        | Type    | Length (bytes) | Alignment | Padding characters | Example  |
|--------------------|---------|----------------|-----------|--------------------|--|
| Product name       | ascii   | 32             | right     | '*'                | *****MyProduct   |
| Tag-ID             | hex     | 8              | right     | '0'                | AB0123F4 or 00000000   |
| Error status       | integer | 1              | right     | '0'                | 0 = No error<br>1 = No tag in association zone<br>2 = More than one tag in association zone<br>3 = No product information available<br>4 = Product information ambiguous<br>5 = Tag battery not OK<br>6 = Tag was recently already associated<br>7 = Product information is not reliable |
| Association status | integer | 1              | fixed     | none               | 0 = disassociated<br>1 = association in progress<br>2 = associated<br>3 = association canceled   |

### Association Example

```
TH0100000160*PRODTAG02*****MainLine*****4711
*****MyProductType*****MyProduct AB0123F402
```

### Disassociation Example

```
TH0100000160*PRODTAG02*****MainLine*****4712
*****MyProductType*****MyProduct000000000
```



**PRODTAG, version 03**

This version uses a hexadecimal representation of an 8-byte tag ID (for a version 3 / D4 tag).

| Description                       | Type    | Length (bytes) | Alignment | Padding characters | Example                                       |
|-----------------------------------|---------|----------------|-----------|--------------------|---|
| <b>Transport Packet (TP)</b>      |         |                |           |                    |   |
| Transport Header                  | ascii   | 2              | fixed     | none               | TH  |
| TP-Version Number                 | integer | 2              | right     | '0'                | 01  |
| TP length                         | integer | 8              | right     | '0'                | 00000168                                      |
| <b>Data Packet (DP)</b>           |         |                |           |                    |   |
| Telegram Type                     | ascii   | 8              | right     | '*'                | *PRODTAG                                      |
| DP-Version Number                 | integer | 2              | right     | '0'                | 03  |
| Tag Circuit                       | ascii   | 32             | right     | '*'                | Name of the tag circuit                       |
| Association / Disassociation Zone | ascii   | 32             | right     | '*'                | Name of the association / disassociation zone |
| Product type information          | ascii   | 32             | right     | '*'                | *****MyProductType                            |

## Telegram Type

| Description        | Type    | Length (bytes) | Alignment | Padding characters | Example  |
|--------------------|---------|----------------|-----------|--------------------|--|
| Product name       | ascii   | 32             | right     | '*'                | *****MyProduct   |
| Tag-ID             | hex     | 16             | right     | '0'                | 0011CE0011223344 or<br>0000000000000000  |
| Error status       | integer | 1              | right     | '0'                | 0 = No error<br>1 = No tag in association zone<br>2 = More than one tag in association zone<br>3 = No product information available<br>4 = Product information ambiguous<br>5 = Tag battery not OK<br>6 = Tag was recently already associated<br>7 = Product information is not reliable |
| Association status | integer | 1              | fixed     | none               | 0 = disassociated<br>1 = association in progress<br>2 = associated<br>3 = association canceled   |

### Association Example

```
TH0100000168*PRODTAG03*****MainLine*****4711
*****MyProductType*****MyProduct 0011CE001122334402
```

### Disassociation Example

```
TH0100000168*PRODTAG03*****MainLine*****4712
*****MyProductType*****MyProduct0000000000000000
```

## SYNCPROD

This telegram type is not implemented in any version of the ACS Protocol service. The telegram definition is a tentative specification for a potential later use.

Information to synchronize product tag associations between systems

This telegram should be used if an external system is the leading system for storing tag product associations. In the case of an ACS system restart, the external system will send all existing tag product associations to ACS.

**Note:** See version 01 below for a variant using 4-byte tag IDs in octet format, version 02 for a variant using a hexadecimal tag ID representation for 4-byte tag IDs, and version 03 for a variant which is capable of transferring 8-byte tag IDs in hexadecimal representation.

If ACS is sending SYNCPROD telegrams, the DP-Version is automatically selected, dependent on the following factors:

- ACS Property 'ACS Protocol Service – UseHexTagID': If set to a value of '0' then DP-Version 01 is always used in SYNCPROD telegrams.
- If a Ubisense Series 7000 location system is in use (both pure and in S7K-D4 co-existence) and if the 'ACS Protocol Service – UseHexTagID' property is set to '1', then DP-Version 02 is used in SYNCPROD telegrams.
- If a Ubisense D4 location system is used (only pure, no S7K-D4 co-existence) and if the 'ACS Protocol Service – UseHexTagID' property is set to '1', then DP-Version 03 is used in SYNCPROD telegrams.

Sender: External system (Server MES or PLC...)

Recipient: ACS

Acknowledgment: yes

SYNCPROD, version 01

| Description   | Type    | Length (bytes) | Alignment | Padding characters | Example   |
|---|---------|----------------|-----------|--------------------|---|
| <b>Transport Packet (TP)</b>                            |         |                |           |                    |   |
| Transport Header  | ascii   | 2              | fixed     | none               | TH  |
| TP-Version Number                                       | integer | 2              | right     | '0'                | 01  |
| TP length   | integer | 8              | right     | '0'                | variable  |
| <b>Data Packet (DP)</b>                                 |         |                |           |                    |   |
| Telegram Type   | ascii   | 8              | right     | '*'                | SYNCPROD  |
| DP-Version Number                                       | integer | 2              | right     | '0'                | 01  |
| Tag Circuit   | ascii   | 32             | right     | '*'                | Name of the tag circuit   |
| Telegram number   | integer | 2              | right     | '0'                | The current telegram number                                       |
| Telegrams total   | integer | 2              | right     | '0'                | The total number of telegrams for one tag circuit synchronization |
| Number of products                                      | integer | 4              | right     | '0'                | 0010  |
| The next fields are repeated 'Number of products' times |         |                |           |                    |   |
| Product type information                                | ascii   | 32             | right     | '*'                | *****4711   |
| Product name  | ascii   | 32             | right     | '*'                | *****MyProduct  |
| Tag-ID  | ascii   | 15             | right     | '*'                | 100-000-032-117   |

**Example**

```
TH0100000133SYNCPROD01*****MainLine01020002*****
****4711*****MyProduct100-000-032-
117*****4711*****OtherProduct100-000-032-118
```

## SYNCPROD, version 02

| Description  | Type    | Length (bytes) | Alignment | Padding characters | Example   |
|--|---------|----------------|-----------|--------------------|---|
| <b>Transport Packet (TP)</b>                             |         |                |           |                    |   |
| Transport Header   | ascii   | 2              | fixed     | none               | TH  |
| TP-Version Number  | integer | 2              | right     | '0'                | 01  |
| TP length  | integer | 8              | right     | '0'                | variable  |
| <b>Data Packet (DP)</b>                                  |         |                |           |                    |   |
| Telegram Type  | ascii   | 8              | right     | '*'                | SYNCPROD  |
| DP-Version Number  | integer | 2              | right     | '0'                | 02  |
| Tag Circuit  | ascii   | 32             | right     | '*'                | Name of the tag circuit   |
| Telegram number  | integer | 2              | right     | '0'                | The current telegram number                                       |
| Telegrams total  | integer | 2              | right     | '0'                | The total number of telegrams for one tag circuit synchronization |
| Number of products                                       | integer | 4              | right     | '0'                | 0010  |
| The next fields are repeated 'Number of products ' times |         |                |           |                    |   |
| Product type information                                 | ascii   | 32             | right     | '*'                | *****4711   |
| Product name   | ascii   | 32             | right     | '*'                | *****MyProduct  |
| Tag-ID   | hex     | 8              | right     | '0'                | 012345AB  |

**Example**

```
TH0100000119SYNCPROD02*****MainLine01020002*****
****4711*****MyProduct012345AB*****4711*****
*****OtherProduct012345CD
```

SYNCPROD, version 03

| Description  | Type    | Length (bytes) | Alignment | Padding characters | Example   |
|--|---------|----------------|-----------|--------------------|---|
| <b>Transport Packet (TP)</b>                             |         |                |           |                    |   |
| Transport Header   | ascii   | 2              | fixed     | none               | TH  |
| TP-Version Number  | integer | 2              | right     | '0'                | 01  |
| TP length  | integer | 8              | right     | '0'                | variable  |
| <b>Data Packet (DP)</b>                                  |         |                |           |                    |   |
| Telegram Type  | ascii   | 8              | right     | '*'                | SYNCPROD  |
| DP-Version Number  | integer | 2              | right     | '0'                | 03  |
| Tag Circuit  | ascii   | 32             | right     | '*'                | Name of the tag circuit   |
| Telegram number  | integer | 2              | right     | '0'                | The current telegram number                                       |
| Telegrams total  | integer | 2              | right     | '0'                | The total number of telegrams for one tag circuit synchronization |
| Number of products                                       | integer | 4              | right     | '0'                | 0010  |
| The next fields are repeated 'Number of products ' times |         |                |           |                    |   |
| Product type information                                 | ascii   | 32             | right     | '*'                | *****4711   |
| Product name   | ascii   | 32             | right     | '*'                | *****MyProduct  |
| Tag-ID   | hex     | 16             | right     | '0'                | 0011CE00012345AB  |

**Example**

```
TH0100000135SYNCPROD03*****MainLine01020002*****
****4711*****MyProduct0011CE00012345AB*****47
11*****OtherProduct0011CE00012345CD
```

## PRODDATA

Information about additional product attributes

This telegram is used either when ACS needs to store additional product attributes (i.e. information on top of the product identifier) or when ACS sends those product attributes to a client (such as a tool controller or a workers assistance system) which needs the attributes in order to do its job (e.g. selecting an appropriate tool program or displaying the attributes).

Sender: ACS or External system (Server MES or PLC...)

Recipient: ACS or External system (PLC, Station PC, Tool controller)

Acknowledgment: yes

| Description                  | Type    | Length (bytes) | Alignment | Padding characters | Example            |
|------------------------------|---------|----------------|-----------|--------------------|--------------------|
| <b>Transport Packet (TP)</b> |         |                |           |                    |                    |
| Transport Header             | ascii   | 2              | fixed     | none               | TH                 |
| TP-Version Number            | integer | 2              | right     | '0'                | 01                 |
| TP length                    | integer | 8              | right     | '0'                | variable           |
| <b>Data Packet (DP)</b>      |         |                |           |                    |                    |
| Telegram Type                | ascii   | 8              | right     | '*'                | PRODDATA           |
| DP-Version Number            | integer | 2              | right     | '0'                | 01                 |
| Product type information     | ascii   | 32             | right     | '*'                | *****MyProductType |
| Product name                 | ascii   | 32             | right     | '*'                | *****MyProduct     |
| Number of attributes         | integer | 4              | right     | '0'                | 0003               |

## Telegram Type

| Description   | Type    | Length (bytes) | Alignment | Padding characters | Example     |
|---|---------|----------------|-----------|--------------------|-------------|
| The next fields are repeated 'Number of attributes' times |         |                |           |                    |             |
| Attribute name  | ascii   | 32             | right     | '*'                | *****Colour |
| Attribute value length                                    | integer | 4              | right     | '0'                | 0003        |
| Attribute value   | ascii   | variable       | right     | none               | red         |

### Example

```
TH0100000129PRODDATA01*****MyProductType*****MyProduct
0001*****Colour0003red
```

### Special Product Attributes

The following attributes have special meanings in ACS:

| Attribute Name | Attribute type and domain | Description   |
|----------------|---------------------------|---|
| SeqNo          | string                    | Product sequence number for products. The sequence number can be displayed in ACS together with the product name (VIN, PNR...). See ACS Help: 'Displaying Sequence Numbers alongside Product Names' |



## PRODDTRQ

Request to send additional product attributes.

This telegram is used to request product attributes, either by ACS or an external system. The other system will send product attributes regarding the requested product using the PRODDATA telegram. The PRODDATA telegram will be sent asynchronously.

Sender: ACS or External system (Server MES or PLC...)

Recipient: ACS or External system (PLC, Station PC, Tool controller)

Acknowledgment: yes

| Description                  | Type    | Length (bytes) | Alignment | Padding characters | Example            |
|------------------------------|---------|----------------|-----------|--------------------|--------------------|
| <b>Transport Packet (TP)</b> |         |                |           |                    |                    |
| Transport Header             | ascii   | 2              | fixed     | none               | TH                 |
| TP-Version Number            | integer | 2              | right     | '0'                | 01                 |
| TP length                    | integer | 8              | right     | '0'                | 00000086           |
| <b>Data Packet (DP)</b>      |         |                |           |                    |                    |
| Telegram Type                | ascii   | 8              | right     | '*'                | PRODDTRQ           |
| DP-Version Number            | integer | 2              | right     | '0'                | 01                 |
| Product type information     | ascii   | 32             | right     | '*'                | *****MyProductType |
| Product name                 | ascii   | 32             | right     | '*'                | *****MyProduct     |

### Example

```
TH0100000086PRODDTRQ01*****MyProductType*****MyProduct
```

### Hint

The 'Product type information' field can be left empty (i.e. it contains only the padding characters). In this case the receiving system will try to find all products with a name as given in 'Product name' and send PRODDATA telegrams for each product. If product names are unique

## Telegram Type

within all product types, the result is that only a single PRODDATA telegram will be sent in response to the PRODDTRQ request.

### **Example**

```
TH010000086PRODDTRQ01*****MyProduct
```

## PRODAT

Telegram containing product location information, e.g. regarding products on an assembly line or products at an association station.

### Use Case 1

By using this telegram ACS can receive the exact position of a product (e.g. on an assembly line) from an external system like a PLC. ACS can use this information to track products on an assembly line as an alternative to using Ubisense tags on the products to track them.

The telegram can also be used at an association station to receive information for products to become associated to tags. In this case only a single product should be submitted and the offset value is not used. If more than one product is transferred, ACS treats this as a conflicting situation to be resolved manually by the user.

Sender: Control system (Server MES or PLC...)

Recipient: ACS

Acknowledgment: yes

### Use Case 2

The telegram can be used by ACS to report when a product has passed an Assembly Line Trigger Point.

Sender: ACS

Recipient: Control system (Server MES or PLC...)

Acknowledgment: yes

| Description                  | Type    | Length (bytes) | Alignment | Padding characters | Example  |
|------------------------------|---------|----------------|-----------|--------------------|----------|
| <b>Transport Packet (TP)</b> |         |                |           |                    |          |
| Transport Header             | ascii   | 2              | fixed     | none               | TH       |
| TP-Version Number            | integer | 2              | right     | '0'                | 01       |
| TP length                    | integer | 8              | right     | '0'                | variable |

## Telegram Type

| Description   | Type    | Length (bytes) | Alignment | Padding characters | Example                                |
|---|---------|----------------|-----------|--------------------|--|
| <b>Data Packet (DP)</b>                                 |         |                |           |                    |  |
| Telegram Type   | ascii   | 8              | right     | '*'                | **PRODAT                               |
| DP-Version Number                                       | integer | 2              | right     | '0'                | 01                                     |
| Location Name   | ascii   | 32             | right     | '*'                | *****Line41<br>*****AssociationStation |
| Number of products                                      | integer | 4              | right     | '0'                | 0010                                   |
| The next fields are repeated 'Number of products' times |         |                |           |                    |  |
| Product type information                                | ascii   | 32             | right     | '*'                | *****4711                              |
| Product name  | ascii   | 32             | right     | '*'                | *****MyProduct                         |
| Offset Value in cm                                      | integer | 6              | right     | '0'                | 006532                                 |

If in a (Product Type, Product Name, Offset) tuple the Product Name is empty, the whole tuple is ignored.

## Examples

### Use Case 1 - Assembly line

```
TH0100000198**PRODAT01*****Line410002*****
4711*****P001006532*****4711*****
*****P002007132
```

### Use Case 1 - Association station

```
TH0100000128**PRODAT01*****AssociationStation0001*****
4711*****MyProduct000000
```

## Externally-driven Assembly Lines

### PLC delivering all products with product offsets

In this case the PLC must send PRODAT telegrams whenever the offset of a single product changes, or at regular time intervals. The PRODAT telegram will contain product information (type, name) for all products on the line, together with a valid offset value for each product.

#### Example

```
TH0100000758**PRODAT01*****460010*****Re
dCar*****46009000300*****GreenCar*****
*****46008000900*****RedCar*****4
6007001500*****GreenCar*****46006002100*****
*****RedCar*****46005002700*****
**GreenCar*****46004003300*****RedCar*****
*****46003003900*****GreenCar*****
*****46002004500*****RedCar*****46001005100
*****GreenCar*****46000005700
```

### PLC delivering all products in a pulsed manner

In this case the PLC must send PRODAT telegrams whenever the sequence of products changes. In this case ACS positions the products at configured distances in the sequence received through the PRODAT telegram. The PRODAT telegram will contain product information (type, name) for all products on the line. The offset value for each product is ignored and can be set to 0. The first product in the telegram will be positioned at the start point of the line, subsequent products towards the end of the assembly line. Gaps can be expressed by a special product type name. The gap type name can be configured in ACS, with a default value of '\_GAP\_'. The product name parameter for gaps is irrelevant and can be set to any name.

## Telegram Type

### Example

```
TH0100000758**PRODAT01*****460010*****_
GAP_
*****GAP000000*****GreenCar*****
*****4600800000*****RedCar*****46007
000000*****GreenCar*****4600600000*****
*****RedCar*****4600500000*****Gr
eenCar*****4600400000*****RedCar*****
*****4600300000*****GreenCar*****
*4600200000*****_GAP_
*****GAP000000*****GreenCar*****
*****4600000000
```

### PLC delivering only the first product in a pulsed manner

In this case the PLC must send PRODAT telegrams whenever the first, i.e. the one at the beginning of the assembly line, changes. The PRODAT telegram will contain product information (type, name) for the very first product on the line. In this case ACS positions all products currently on the line one slot further towards the end of the line, and the product contained in the telegram at the assembly line start. The offset value for the product is ignored and can be set to 0. Gaps can be expressed by a special product type name. The gap type name can be configured in ACS, with a default value of '\_GAP\_'. The product name parameter for gaps is usually irrelevant and can be set to any name. However, if there are subsequent gaps their names must be set to unique values.

### Example

```
TH0100000128**PRODAT01*****460001*****Re
dCar*****RC01000000

TH0100000128**PRODAT01*****460001*****Gree
nCar*****GC01000000
TH0100000128**PRODAT01*****460001*****_
GAP_*****GAP000000

TH0100000128**PRODAT01*****460001*****Re
dCar*****RC02000000
TH0100000128**PRODAT01*****460001*****_
GAP_*****Gap1000000
TH0100000128**PRODAT01*****460001*****_
GAP_*****GAP2000000

TH0100000128**PRODAT01*****460001*****Gree
nCar*****GC02000000
```

### Trigger point telegram example

```
TH0100000128**PRODAT01*****TP-40-
15.50001*****VEHICLE*****CAR-001001550
```

## ISINZ

Telegram Type indicating that an object is contained in a zone.

The telegram is sent when the object enters a zone.

The object can be a product or a device.

This telegram is used to inform an external system like a tool controller or a PLC that either a product (e.g. a car) has entered a certain zone (e.g. a checkpoint on an assembly line or a space in front of a test station), or that a device (e.g. a tool) has entered a certain zone such as a station on an assembly line or a zone of a product (e.g. the zone around the engine bay of a car).

There are two versions of ISINZ.

- Version 1: This telegram provides the container name and the name of the contained object as parameters. It is used
  - If a device enters a static space (ObjectIsLocatedEvent)
  - If a product enters a static space (ContainerContainsObjectEvent, ProductIsLocatedEvent)
- Version 2: This telegram provides the container name, the container space name and the name of the contained object as parameters. It is used
  - If a device enters a product space (ObjectIsLocatedEvent)

## Telegram Type

### ISINZ Version 01

Sender: ACS

Recipient: External system (Server MES, PLC, station PC, tool controller...)

Acknowledgment: yes

| Description                  | Type    | Length (bytes) | Alignment | Padding characters | Example                  |
|------------------------------|---------|----------------|-----------|--------------------|--------------------------|
| <b>Transport Packet (TP)</b> |         |                |           |                    |                          |
| Transport Header             | ascii   | 2              | fixed     | none               | TH                       |
| TP-Version Number            | integer | 2              | right     | '0'                | 01                       |
| TP length                    | integer | 8              | right     | '0'                | 00000086                 |
| <b>Data Packet (DP)</b>      |         |                |           |                    |                          |
| Telegram Type                | ascii   | 8              | right     | '*'                | ***ISINZ                 |
| DP-Version Number            | integer | 2              | right     | '0'                | 01                       |
| Container name               | ascii   | 32             | right     | '*'                | *****ContainerName       |
| Object name                  | ascii   | 32             | right     | '*'                | *****ContainedObjectName |

### Example

```
TH010000086***ISINZ01*****ContainerName*****ContainedObjectName
```



**ISINZ Version 02**

Sender: ACS

Recipient: External system (Server MES, PLC, station PC, tool controller...)

Acknowledgment: yes

| Description                  | Type    | Length (bytes) | Alignment | Padding characters | Example                  |
|------------------------------|---------|----------------|-----------|--------------------|--------------------------|
| <b>Transport Packet (TP)</b> |         |                |           |                    |                          |
| Transport Header             | ascii   | 2              | fixed     | none               | TH                       |
| TP-Version Number            | integer | 2              | right     | '0'                | 01                       |
| TP length                    | integer | 8              | right     | '0'                | 00000112                 |
| <b>Data Packet (DP)</b>      |         |                |           |                    |                          |
| Telegram Type                | ascii   | 8              | right     | '*'                | ***ISINZ                 |
| DP-Version Number            | integer | 2              | right     | '0'                | 02                       |
| Container name               | ascii   | 32             | right     | '*'                | *****ContainerName       |
| Container space name         | ascii   | 32             | right     | '*'                | *****ContainerSpaceName  |
| Object                       | ascii   | 32             | right     | '*'                | *****ContainedObjectName |

**Example**

```
TH0100000118***ISINZ02*****ContainerName*****ContainerSpaceName
*****ContainedObjectName
```

Telegram Type

## ISOUTZ

Telegram Type indicating that an object is contained in a zone.

The telegram is sent when the object leaves a zone.

See the description of the [IZINZ telegram](#) above for information concerning the usage of this telegram.

There are two versions for ISOUTZ.

- Version 1: This telegram provides the container name and the name of the contained object as parameters. It is used
  - If a device leaves a static space (ObjectIsLocatedEvent)
  - If a product leaves a static space (ContainerContainsObjectEvent, ProductIsLocatedEvent)
- Version 2: This telegram provides the container name, the container space name and the name of the contained object as parameters. It is used
  - If a device leaves a product space (ObjectIsLocatedEvent)

**ISOUTZ Version 01**

Sender: ACS

Recipient: External system (Server MES, PLC, station PC, tool controller...)

Acknowledgment: yes

| Description                  | Type    | Length (bytes) | Alignment | Padding characters | Example                  |
|------------------------------|---------|----------------|-----------|--------------------|--------------------------|
| <b>Transport Packet (TP)</b> |         |                |           |                    |                          |
| Transport Header             | ascii   | 2              | fixed     | none               | TH                       |
| TP-Version Number            | integer | 2              | right     | '0'                | 01                       |
| TP length                    | integer | 8              | right     | '0'                | 00000086                 |
| <b>Data Packet (DP)</b>      |         |                |           |                    |                          |
| Telegram Type                | ascii   | 8              | right     | '*'                | **ISOUTZ                 |
| DP-Version Number            | integer | 2              | right     | '0'                | 01                       |
| Container space name         | ascii   | 32             | right     | '*'                | *****ContainerSpaceName  |
| Object                       | ascii   | 32             | right     | '*'                | *****ContainedObjectName |

**Example**

```
TH0100000086**ISOUTZ01*****ContainerSpaceName*****ContainedObjectName
```

## Telegram Type

### ISOUTZ Version 02

Sender: ACS

Recipient: External system (Server MES, PLC, station PC, tool controller...)

Acknowledgment: yes

| Description                  | Type    | Length (bytes) | Alignment | Padding characters | Example                  |
|------------------------------|---------|----------------|-----------|--------------------|--------------------------|
| <b>Transport Packet (TP)</b> |         |                |           |                    |                          |
| Transport Header             | ascii   | 2              | fixed     | none               | TH                       |
| TP-Version Number            | integer | 2              | right     | '0'                | 01                       |
| TP length                    | integer | 8              | right     | '0'                | 00000118                 |
| <b>Data Packet (DP)</b>      |         |                |           |                    |                          |
| Telegram Type                | ascii   | 8              | right     | '*'                | **ISOUTZ                 |
| DP-Version Number            | integer | 2              | right     | '0'                | 02                       |
| Container name               | ascii   | 32             | right     | '*'                | *****ContainerName       |
| Container space name         | ascii   | 32             | right     | '*'                | *****ContainerSpaceName  |
| Object                       | ascii   | 32             | right     | '*'                | *****ContainedObjectName |

### Example

```
TH0100000118**ISOUTZ02*****ContainerName*****ContainerSpaceName
*****ContainedObjectName
```

## ZONESTAT

Telegram type either describing which objects are currently contained in a zone or describing which zones an object currently is in.

The telegram is sent whenever an object enters or leaves a zone.

The object can be a product or a device.

This telegram is used to inform an external system such as a tool controller or a PLC that either a product (e.g. a car) has entered a certain zone (e.g. a checkpoint on an assembly line or a space in front of a test station) or that a device (e.g. a tool) has entered a certain zone like a station on an assembly line or a zone of a product (e.g. the zone around the engine bay of a car).

This telegram is useful if the object can be contained within more than one zone at the same time.

There are three versions of ZONESTAT.

- Version 1: This telegram provides the container name and a list of names of contained objects as parameters. It is used
  - If a product enters or leaves a static space (ContainerContainsObjectEvent)
- Version 2: This telegram provides the name of the contained object and a list of pairs of container object names and container space names as parameters. It is used
  - If a device enters or leaves a product space (ObjectIsLocatedEvent)
- Version 3: This telegram provides the name of the contained object and a list of names of container objects as parameters. It is used
  - If a device enters or leaves a static space and no product space activation is configured (ObjectIsLocatedEvent)

The ACS Protocol implementation of ACS allows the use of either ZONESTAT or ISINZ/ISOUTZ, but not both. This can be configured per external system.

## Telegram Type

### ZONESTAT Version 01

Telegram type describing which objects are currently contained in a zone. ZONESTAT version 01 is used when a container, e.g. an Ident Zone, Workspace or Station, owns only a single space and can contain multiple objects at the same time.

Sender: ACS

Recipient: External system (Server MES, PLC, station PC, tool controller...)

Acknowledgment: yes

| Description  | Type    | Length (bytes) | Alignment | Padding characters | Example                  |
|--|---------|----------------|-----------|--------------------|--------------------------|
| <b>Transport Packet (TP)</b>                                   |         |                |           |                    |                          |
| Transport Header   | ascii   | 2              | fixed     | none               | TH                       |
| TP-Version Number  | integer | 2              | right     | '0'                | 01                       |
| TP length  | integer | 8              | right     | '0'                | 00000086                 |
| <b>Data Packet (DP)</b>  |         |                |           |                    |                          |
| Telegram Type  | ascii   | 8              | right     | '*'                | ZONESTAT                 |
| DP-Version Number  | integer | 2              | right     | '0'                | 01                       |
| Container space name   | ascii   | 32             | right     | '*'                | *****ContainerSpaceName  |
| Number of contained objects                                    | integer | 4              | right     | '0'                | 0002                     |
| The next field is repeated 'Number of contained objects' times |         |                |           |                    |                          |
| Contained Object   | ascii   | 32             | right     | '*'                | *****ContainedObjectName |

If the contained objects are products on an assembly line, then ACS delivers the list of Contained Objects sorted by their order on the assembly line, products with the lowest distance from the assembly line start point coming first. This includes shadow products, if configured.

**Example**

```
TH0100000122ZONESTAT01*****ContainerSpaceName0002*****ContainedObjectName1*****ContainedObjectName2
```

```
TH0100000122ZONESTAT01*****IdentZone47110002*****Car1*****Car2
```

**ZONESTAT Version 02**

Telegram type describing which zones an object currently is in. ZONESTAT version 02 is used in cases where the contained object, e.g. a device, can be contained in multiple zones at the same time, where the zones can be owned by the same container object, such as multiple, overlapping product spaces of the same product.

Sender: ACS

Recipient: External system (Server MES, PLC, station PC, tool controller...)

Acknowledgment: yes

| Description                  | Type    | Length (bytes) | Alignment | Padding characters | Example                  |
|------------------------------|---------|----------------|-----------|--------------------|--------------------------|
| <b>Transport Packet (TP)</b> |         |                |           |                    |                          |
| Transport Header             | ascii   | 2              | fixed     | none               | TH                       |
| TP-Version Number            | integer | 2              | right     | '0'                | 01                       |
| TP length                    | integer | 8              | right     | '0'                | 00000086                 |
| <b>Data Packet (DP)</b>      |         |                |           |                    |                          |
| Telegram Type                | ascii   | 8              | right     | '*'                | ZONESTAT                 |
| DP-Version Number            | integer | 2              | right     | '0'                | 02                       |
| Contained Object             | ascii   | 32             | right     | '*'                | *****ContainedObjectName |

## Telegram Type

| Description  | Type    | Length (bytes) | Alignment | Padding characters | Example                 |
|--|---------|----------------|-----------|--------------------|-------------------------|
| Number of zones                                      | integer | 4              | right     | '0'                | 0002                    |
| The next fields are repeated 'Number of zones' times |         |                |           |                    |                         |
| Container name                                       | ascii   | 32             | right     | '*'                | *****ContainerName      |
| Container space name                                 | ascii   | 32             | right     | '*'                | *****ContainerSpaceName |

### Example

```
TH0100000186ZONESTAT02*****ContainedObjectName0002*****ContainerName1*****ContainerSpaceName1*****ContainerName1*****ContainerSpaceName2
```

```
TH0100000186ZONESTAT02*****Device47110002*****Car99*****FrontLeft*****Car99*****FrontRight
```

### ZONESTAT Version 03

Telegram type describing which zones an object currently is in. ZONESTAT version 03 is used in cases where the contained object, e.g. a device, can be contained in multiple zones, e.g. stations or workspaces, at the same time.

Sender: ACS

Recipient: External system (Server MES, PLC, station PC, tool controller...)

Acknowledgment: yes

| Description                  | Type    | Length (bytes) | Alignment | Padding characters | Example  |
|------------------------------|---------|----------------|-----------|--------------------|----------|
| <b>Transport Packet (TP)</b> |         |                |           |                    |          |
| Transport Header             | ascii   | 2              | fixed     | none               | TH       |
| TP-Version Number            | integer | 2              | right     | '0'                | 01       |
| TP length                    | integer | 8              | right     | '0'                | 00000086 |



| Description   | Type    | Length (bytes) | Alignment | Padding characters | Example                  |
|---|---------|----------------|-----------|--------------------|--------------------------|
| <b>Data Packet (DP)</b>                                 |         |                |           |                    |                          |
| Telegram Type   | ascii   | 8              | right     | '*'                | ZONESTAT                 |
| DP-Version Number                                       | integer | 2              | right     | '0'                | 03                       |
| Contained Object  | ascii   | 32             | right     | '*'                | *****ContainedObjectName |
| Number of containers                                    | integer | 4              | right     | '0'                | 0002                     |
| The next field is repeated 'Number of containers' times |         |                |           |                    |                          |
| Container name  | ascii   | 32             | right     | '*'                | *****ContainerName       |

### Example

```
TH0100000122ZONESTAT03*****ContainedObjectName0002*****ContainerSpaceName1*****ContainerSpaceName2
```

```
TH0100000122ZONESTAT03*****Device47110002*****Workspace1*****Workspace2
```

## LINESTAT

The LINESTAT telegram is used to deliver the current status of an assembly line, including general information about line movements or current error values and specific information for each product currently tracked by the assembly line.

The telegram is sent whenever the assembly line status changes, as evaluated by the ACS Tracking Support services.

Sender: ACS

Recipient: External system (Server MES, PLC...)

Acknowledgment: no

## Telegram Type

| Description                  | Type    | Length (bytes) | Alignment | Padding characters | Example  |
|------------------------------|---------|----------------|-----------|--------------------|--|
| <b>Transport Packet (TP)</b> |         |                |           |                    |  |
| Transport Header             | ascii   | 2              | fixed     | none               | TH   |
| TP-Version Number            | integer | 2              | right     | '0'                | 01   |
| TP length                    | integer | 8              | right     | '0'                | 00000152<br>=00000080+N*72   |
| <b>Data Packet (DP)</b>      |         |                |           |                    |  |
| Telegram Type                | ascii   | 8              | right     | '*'                | LINESTAT   |
| DP-Version Number            | integer | 2              | right     | '0'                | 01   |
| Telegram number              | i       | 4              | R         | ,0'                | 0013<br><br>The telegram number starts with '0000' for the first telegram after the connection is established and is then incremented by 1 for each new telegram. The next number after '9999' is '0000' |
| Line name                    | a       | 32             | R         | ,*'                | *****Line-50   |
| Line is in motion            | i       | 1              | n/a       | n/a                | 0=no, 1=yes  |
| Tracking is reliable         | i       | 1              | n/a       | n/a                | 0=no, 1=yes  |
| Total error                  | i       | 8              | R         | ,0'                | The sum of all absolute values of errors  product offset – tag offset  [cm].   |

| Description   | Type | Length (bytes) | Alignment | Padding characters | Example   |
|---|------|----------------|-----------|--------------------|---|
| Cumulated error   | i    | 8              | R         | ,0'                | The sum of all values of errors (product offset – tag offset) [cm]        |
| Number of contained information tuples                                      | i    | 4              | R         | ,0'                | 0024<br>The next fields are repeated 'Number of contained tuples' times   |
| The next fields are repeated 'Number of contained information tuples' times |      |                |           |                    |   |
| Contained Object  | a    | 32             | R         | ,*'                | *****ContainedObjectName  |
| Station   | a    | 32             | R         | ,*'                | *****Station-1  |
| Offset  | i    | 8              | R         | ,0'                | The product position along the line (distance from line start point) [cm] |

### Example

```
TH0100000368LINESTAT010013*****Line501100000453000000920004*****
*****KENNR1*****Station100002223*****
*****KENNR2*****Station200001623*****SP-50-
01*****Station300001023*****KENNR4*****
*****Station400000423
```

Contained Object is the name of a product on the assembly line. This can also be a shadow product, if the ACS production line is configured for shadow products. Shadow products represent gaps between real products on the assembly line.

The Station delivered is the one in which the product origin lies. If station extents overlap and multiple stations are found only the first one found is delivered.

There are two ACS global configuration parameters which can be used to filter the stations delivered by LINESTAT:

- 'ACS Protocol Service' – 'Exclude Stations With Description': If non empty, don't use stations with a description including the given identifier in LINESTAT telegrams
- 'ACS Protocol Service' – 'Include Stations With Description': If non empty, only use stations with a description including the given identifier in LINESTAT telegrams

## Telegram Type

The Offset value is the distance of a product position from the start point of the assembly line. The unit of measure is cm.

ACS delivers the list of Contained Objects (products) sorted by their order on the assembly line, products with the lowest distance from the assembly line start point coming first. This includes shadow products, if configured.

## IENABLE

This telegram type is not implemented in any version of the ACS Protocol service. The telegram definition is a tentative specification for a potential later use.

Telegram indicating whether or not a device is enabled.

The telegram is sent when the status changes, as evaluated by the ACS location rules.

Sender: ACS

Recipient: External system (Server MES, PLC, station PC, tool controller...)

Acknowledgment: yes

| Description                  | Type    | Length (bytes) | Alignment | Padding characters | Example               |
|------------------------------|---------|----------------|-----------|--------------------|-----------------------|
| <b>Transport Packet (TP)</b> |         |                |           |                    |                       |
| Transport Header             | ascii   | 2              | fixed     | none               | TH                    |
| TP-Version Number            | integer | 2              | right     | '0'                | 01                    |
| TP length                    | integer | 8              | right     | '0'                | 00000055              |
| <b>Data Packet (DP)</b>      |         |                |           |                    |                       |
| Telegram Type                | ascii   | 8              | right     | '*'                | IENABLE               |
| DP-Version Number            | integer | 2              | right     | '0'                | 01                    |
| Device name                  | ascii   | 32             | right     | '*'                | *****DeviceName       |
| Is enabled                   | integer | 1              | fixed     | none               | 1 = true<br>0 = false |

### Example

```
TH0100000055IENABLE01*****DeviceName1
TH0100000055IENABLE01*****DeviceName0
```

## SUBSCR

Subscribe to events.

An external system can subscribe to receiving events generated by an object entering or leaving a zone, by specifying a number of subscription parameters. After a successful subscription, the external system will receive ISINZ and ISOUTZ or ZONESTAT telegrams, whenever corresponding objects enter or leave zones configured by the subscription.

Configuration parameters are:

- Subscription Name
- Event type
- Container Names
- Container Types
- Container Space Names
- Contained Object Names
- Contained Object Types

Sender: External system (Server MES, PLC, station PC, tool controller...)

Recipient: ACS

Acknowledgment: yes

| Description                  | Type    | Length (bytes) | Alignment | Padding characters | Example  |
|------------------------------|---------|----------------|-----------|--------------------|----------|
| <b>Transport Packet (TP)</b> |         |                |           |                    |          |
| Transport Header             | ascii   | 2              | fixed     | none               | TH       |
| TP-Version Number            | integer | 2              | right     | '0'                | 01       |
| TP length                    | integer | 8              | right     | '0'                | variable |
| <b>Data Packet (DP)</b>      |         |                |           |                    |          |
| Telegram Type                | ascii   | 8              | right     | '**'               | **SUBSCR |
| DP-Version Number            | integer | 2              | right     | '0'                | 01       |

| Description   | Type    | Length (bytes) | Alignment | Padding characters | Example   |
|---|---------|----------------|-----------|--------------------|---|
| Subscription name   | ascii   | 32             | right     | '*'                | *****SomeSubscriptionID   |
| Event type  | integer | 2              | right     | '0'                | 01 = ObjectIsLocatedEvent<br>02 = ContainerContainsObjectEvent<br>03 = ProductIsLocatedEvent<br>04 =<br>ProductHasPassedTriggerPointEvent |
| Container objects configuration count   | integer | 4              | right     | '0'                | 0001  |
| The next three fields (Container name, type, space names) are repeated 'Container objects configuration count' times          |         |                |           |                    |   |
| Container name  | ascii   | 32             | right     | '*'                | *****SomePNR  |
| Container type  | ascii   | 32             | right     | '*'                | *****Vehicle  |
| Container space name  | ascii   | 32             | right     | '*'                | *****Bonnet   |
| Contained objects configuration count   | integer | 4              | right     | '0'                | 0001  |
| The next two fields (Contained object name, Contained object type) are repeated 'Contained objects configuration count' times |         |                |           |                    |   |
| Contained object name   | ascii   | 32             | right     | '*'                | *****Device-0001  |
| Contained object type   | ascii   | 32             | right     | '*'                | *****Some Device Type   |

A subscription can be changed by sending another SUBSCR telegram with the same Subscription Name as before, with changed parameters.

There are restrictions in the current implementation:

## Telegram Type

- The only event type supported is 01 = ObjectIsLocatedEvent
- Containers must be ACS Products. Other container types (ACS Station, ACS Workspace, ACS Ident Zone) are currently not supported
- If no container types (ACS Product Types) are configured, all existing product types will be used
- Contained objects are ACS Devices
- The number of contained object names must be 1
- If no contained object type is given and the given 'Contained object name' is not unique in ACS an error is returned

If a subscription can be made, the RESPONSE returned is OK.

If the configuration parameters of a subscription are erroneous, the RESPONSE will be 'Not OK' and the following error codes can be returned. If there are multiple errors only the first error code will be used.

| Error code | Error description             |
|------------|-------------------------------|
| 0100       | Unspecific error              |
| 0101       | Unsupported event type        |
| 0102       | Invalid subscription name     |
| 0103       | Internal error                |
| 0111       | Container instance unknown    |
| 0112       | Container type unknown        |
| 0113       | Container type invalid        |
| 0114       | Container space unknown       |
| 0121       | Contained instance unknown    |
| 0122       | Contained type unknown        |
| 0123       | Contained type invalid        |
| 0124       | Contained instance not unique |

### Example

Subscribe for an ObjectIsLocated event for the device 'MyDevice' of type 'PowerFocus', to be activated in product zone 'Bonnet' for all products of type 'Vehicle'



```
TH010000224**SUBSCR01*****MyDeviceSubscription010001*****
*****Vehicle*****Bonnet0001*****
*****MyDevice*****PowerFocus
```

Subscribe for an ObjectIsLocated event for the device 'MyDevice', to be activated in product zone 'Bonnet' for a product of type 'Vehicle' with ID '12345678'

```
TH010000224**SUBSCR01*****MyDeviceSubscription010001*****12
345678*****Vehicle*****Bonnet0001*****
*****MyDevice*****
```

### ACS configuration hint

The Additional Parameters of external systems running ACS Protocol version 1.0.0 provide the parameter 'Endpoint Roles'. There is a role 'SubscriptionHandler', which can be configured. An external system having the role 'SubscriptionHandler' will only send device events (ObjectIsLoatedEvents), if the connected system has subscribed for events. i.e.

- if the connected system has sent a SUBSCR telegram for an associated device, the activation rules of the device will be adapted and events will be forwarded (ISINZ/ISOUTZ or ZONESTAT, depending on the 'Send ZONESTAT' additional parameter)
- if the connected system sends an UNSUBSCR, device events will no longer be forwarded.

In case of receiving a SUBSCR telegram the rules included in the SUBSCR telegram will be added to the device rules, if necessary, but existing rules will not be removed. This is to avoid frequent rule changes. The ACS Protocol will then filter out activation events which do not comply with the rules in the SUBSCR telegram.

This behavior can be overridden by setting the global ACS parameter 'ACS Protocol Service' - 'Store subscription rules persistently' to 'true'. For testing purposes, when using the 'Device Monitoring' ACS page, this might be handy. However, if the subscriptions change frequently, e.g. for every product, the parameter should be set to 'false'

Telegram Type

## UNSUBSCR

Unsubscribe events.

An external system can retract a subscription for receiving events, which was initiated before.

Configuration parameters are

- Subscription Name

Sender: External system (Server MES, PLC, station PC, tool controller...)

Recipient: ACS

Acknowledgment: yes

| Description                  | Type    | Length (bytes) | Alignment | Padding characters | Example                 |
|------------------------------|---------|----------------|-----------|--------------------|-------------------------|
| <b>Transport Packet (TP)</b> |         |                |           |                    |                         |
| Transport Header             | ascii   | 2              | fixed     | none               | TH                      |
| TP-Version Number            | integer | 2              | right     | '0'                | 01                      |
| TP length                    | integer | 8              | right     | '0'                | variable                |
| <b>Data Packet (DP)</b>      |         |                |           |                    |                         |
| Telegram Type                | ascii   | 8              | right     | **                 | UNSUBSCR                |
| DP-Version Number            | integer | 2              | right     | '0'                | 01                      |
| Subscription name            | ascii   | 32             | right     | **                 | *****SomeSubscriptionID |

The subscription for the given name will be removed.

If the subscription name is left empty (contains only padding chars), all subscriptions referring to the external system will be removed.

### Example

Unsubscribe from a formerly made subscription with the name 'MyDeviceSubscription'

```
TH0100000054UNSUBSCR01*****MyDeviceSubscription
```

Unsubscribe from all formerly made subscriptions

TH010000054UNSUBSCR01\*\*\*\*\*

## Telegram Type

### EXCEPT

Exception Telegram.

This telegram is sent when an exception occurs in ACS.

After the connection is established exceptions that occurred while the connection was down will be sent.

Sender: ACS

Recipient: External system (Server MES, PLC, issue tracking system, ...)

Acknowledgment: yes

| Description                  | Type    | Length (bytes)                 | Alignment | Padding characters | Example                    |
|------------------------------|---------|--------------------------------|-----------|--------------------|----------------------------|
| <b>Transport Packet (TP)</b> |         |                                |           |                    |                            |
| Transport Header             | ascii   | 2                              | fixed     | none               | TH                         |
| TP-Version Number            | integer | 2                              | right     | '0'                | 01                         |
| TP length                    | integer | 8                              | right     | '0'                | 0000022                    |
| <b>Data Packet (DP)</b>      |         |                                |           |                    |                            |
| Telegram Type                | ascii   | 8                              | right     | '*'                | **EXCEPT                   |
| DP-Version Number            | integer | 2                              | right     | '0'                | 01                         |
| Severity                     | integer | 1                              | fixed     | none               | 3                          |
| Exception Code               | integer | 5                              | right     | '0'                | 00174                      |
| Length of description        | integer | 10                             | right     | '0'                | 000000011                  |
| Description Key              | ascii   | variable, to a maximum of 128  | fixed     | none               | Example_Key                |
| Length of information        | integer | 10                             | right     | '0'                | 000000017                  |
| Information Key              | ascii   | variable, to a maximum of 1024 | fixed     | none               | info1 info2 info3          |
| Time                         | ascii   | 23                             | fixed     | none               | MM/dd/YYYY<br>HH:mm:ss.fff |

**Severity**

Levels of severity are:

0 = None

1 = Information

2 = Warning

3 = Error

**Example**

```
TH0100000099**EXCEPT013001740000000011Example_Key0000000017info1|info2|info308/24/2013
15:14:53.250
```

**Exception Codes and Messages**

For a full list of exception codes and messages, see [ACS\\_Exceptions.pdf](#) on the [Ubisense Documentation Portal](#).

In the list of exception messages, the numbers within curly braces refer to the zero-based index in the Information Key, e.g.

**Exception Message**

```
TH0100000112**EXCEPT0110002100000000000000000041Tracking Support|M021_LC_
STR1|Running|ACS12/10/2019 11:38:41.276
```

**Exception Information Key**

```
Tracking Support|M021_LC_STR1|Running|ACS
```

**Message Text**

```
Service {3}-{0} [{1}] is running
```

**Filled-out message text**

```
Service ACS-Tracking Support [M021_LC_STR1] is running
```

**Example Messages****Battery condition is ok**

```
TH0100000101**EXCEPT0110000100000000000000000030017-206-000-000-000-000-
001|OK01/12/2021 10:38:19.700
```

**Battery condition is warning**

## Telegram Type

TH0100000106\*\*EXCEPT0120000200000000000000000000000035017-206-000-000-000-000-001|WARNING01/12/2021 10:38:21.128

### **Battery condition is failing**

TH0100000106\*\*EXCEPT0120000300000000000000000000000035017-206-000-000-000-000-001|FAILING01/12/2021 10:38:22.533

### **Sensor is not running**

TH0100000147\*\*EXCEPT01200036000000000000000000000007600:11:CE:00:07:CD|00:11:CE:00:07:CD|10.49.2.123|NotRunning|31001|Master|True01/12/2021 10:35:44.814

### **Service is stopped**

TH0100000120\*\*EXCEPT01200023000000000000000000000049Ident Determination|M050B36\_GC\_35\_36|Ubisense|ACS01/12/2021 10:31:15.253

## EXCEPTRQ

Exception Request Telegram.

This telegram can be used to request specific former EXCEPT messages again. It allows you to configure which messages are sent again.

Sender: External system (Server MES, PLC, issue tracking system, ...)

Recipient: ACS

Acknowledgment: no

| Description                  | Type    | Length (bytes) | Alignment | Padding characters | Example       |
|------------------------------|---------|----------------|-----------|--------------------|---------------|
| <b>Transport Packet (TP)</b> |         |                |           |                    |               |
| Transport Header             | ascii   | 2              | fixed     | none               | TH            |
| TP-Version Number            | integer | 2              | right     | '0'                | 01            |
| TP length                    | integer | 8              | right     | '0'                | 0000029       |
| <b>Data Packet (DP)</b>      |         |                |           |                    |               |
| Telegram Type                | ascii   | 8              | right     | '*'                | EXCEPTRQ      |
| DP-Version Number            | integer | 2              | right     | '0'                | 01            |
| Group                        | integer | 2              | right     | '0'                | 01            |
| SeverityThreshold            | integer | 1              | fixed     | none               | 2             |
| Length of ID Field           | integer | 4              | right     | '0'                | 0013          |
| ID Field                     | ascii   | variable       | fixed     | none               | SHAGAT3860113 |

Group: The Group field allows you to select the type of information requested. The following values are supported:

- 1 = Sensors
- 2 = Tags
- 3 = Services

SeverityThreshold: The SeverityThreshold allows the selection of the minimum severity for the requested information. Only EXCEPT telegrams with severity equal to or above the requested severity will be returned. The following values are supported:

- 0 = None

## Telegram Type

1 = Information

2 = Warning

3 = Error

ID Field: The ID field allows the selection of specific entities, or it can be left empty for information on all entities to be returned. The following IDs are supported:

- Sensors: Name, MAC address in HEX, with colon char between bytes, e.g. '00:11:CE:00:1F:4D'
- Tags: ID, either in octet format, e.g. '251-000-112-049' or in HEX, e.g. '6400127A' or '64:00:12:7A'
- Services: Colon separated ID, consisting of Package Name:Service Name:Cell Name, e.g. 'ACS:Tracking Support:M021\_LC\_STR1'

### Sensor query examples

Query the status of all sensors:

```
TH0100000029EXCEPTRQ010110000
```

Query the status of all sensors, with a minimum exception severity of 'Warning':

```
TH0100000029EXCEPTRQ010120000
```

Query the status of sensor 00:11:CE:00:84:83:

```
TH0100000046EXCEPTRQ01010001700:11:CE:00:84:83
```

Query the status of sensor 00:11:CE:00:84:83, by name:

```
TH0100000049EXCEPTRQ010100020M022ACS010812 :84:83
```

### Tag query examples

Query the status of all tags:

```
TH0100000029EXCEPTRQ010200000
```

Query the status of all tags, with a minimum exception severity of 'Warning':

```
TH0100000029EXCEPTRQ010220000
```

Query the status of tag 32:00:05:8F

```
TH0100000037EXCEPTRQ0102000083200058F
```



Query the status of tag 32:00:05:8F, alternative

```
TH0100000040EXCEPTRQ01020001132:00:05:8F
```

### Service query examples

Query the status of all services:

```
TH0100000029EXCEPTRQ010300000
```

Query the status of all services, with a minimum exception severity of 'Warning':

```
TH0100000029EXCEPTRQ010320000
```

Query the status of service 'Events', running on cell 'Site', package 'UTCP':

```
TH0100000045EXCEPTRQ010300016UTCP:Events:Site
```

Query the status of service 'Tracking Support', running on cell 'M021\_LC\_STR1', package 'ACS':

```
TH0100000062EXCEPTRQ010300033ACS:Tracking Support:M021_LC_STR1
```

## EXCEPTRS

Response to Exception Request Telegram. This telegram indicates how many EXCEPT telegrams can be expected as a result of an EXCEPTRQ telegram.

Sender: ACS

Recipient: External system (Server MES, PLC, issue tracking system, ...)

Acknowledgment: no

| Description                  | Type    | Length (bytes) | Alignment | Padding characters | Example    |
|------------------------------|---------|----------------|-----------|--------------------|------------|
| <b>Transport Packet (TP)</b> |         |                |           |                    |            |
| Transport Header             | ascii   | 2              | fixed     | none               | TH         |
| TP-Version Number            | integer | 2              | right     | '0'                | 01         |
| TP length                    | integer | 8              | right     | '0'                | 0000034    |
| <b>Data Packet (DP)</b>      |         |                |           |                    |            |
| Telegram Type                | ascii   | 8              | right     | '*                 | EXCEPTRS   |
| DP-Version Number            | integer | 2              | right     | '0'                | 01         |
| Group                        | integer | 2              | right     | '0'                | 01         |
| Number of EXCEPT telegrams   | integer | 10             | right     | '0'                | 0000000001 |

Group: The Group field mirrors the value of the Group field as received in the EXCEPTRQ telegram.

Number of EXCEPT telegrams: This is the number of EXCEPT telegrams which will follow as a result of the EXCEPTRQ telegram. This number can be zero, if the query does not return entities matching the query criteria. This will be the case for example, if the EXCEPTRQ telegram specified the query as 'all sensors with severity greater equal 2' and all sensors are up and running.

### Example

```
TH0100000034EXCEPTRS01010000000002
```

## TAGPOS

Telegram for sending tag position information.

This telegram has been used within the ACSsupport package for transferring live tag position data from a production site to a test site (Location Transfer Service).

Sender: ACS

Recipient: External system (Server MES, PLC...)

Acknowledgment: no

### TAGPOS Version 01

| Description  | Type    | Length (bytes) | Alignment | Padding characters | Example                                 |
|--|---------|----------------|-----------|--------------------|---|
| <b>Transport Packet (TP)</b>                             |         |                |           |                    |   |
| Transport Header   | ascii   | 2              | fixed     | none               | TH                                      |
| TP-Version Number  | integer | 2              | right     | '0'                | 01                                      |
| TP length  | integer | 8              | right     | '0'                | variable                                |
| <b>Data Packet (DP)</b>                                  |         |                |           |                    |   |
| Telegram Type  | ascii   | 8              | right     | **'                | **TAGPOS                                |
| DP-Version Number  | integer | 2              | right     | '0'                | 01                                      |
| Number of positions                                      | hex     | 4              | right     | '0'                | 000A                                    |
| The next fields are repeated 'Number of positions' times |         |                |           |                    |   |
| Tag-ID   | hex     | 8              | right     | '0'                | 6400011F                                |
| X coordinate   | hex     | 8              | right     | '0'                | 00002710<br>Value in cm (here 10000 cm) |
| Y coordinate   | hex     | 8              | right     | '0'                | Value in cm                             |
| Z coordinate   | hex     | 8              | right     | '0'                | Value in cm                             |

### Example

## Telegram Type

TH0100000026\*\*TAGPOS010000

TH0100000058\*\*TAGPOS01000164000001000027100000271000000001

## TAGPOS Version 02

| Description  | Type    | Length (bytes) | Alignment | Padding characters | Example                                |
|--|---------|----------------|-----------|--------------------|--|
| <b>Transport Packet (TP)</b>                             |         |                |           |                    |  |
| Transport Header   | ascii   | 2              | fixed     | none               | TH                                     |
| TP-Version Number  | integer | 2              | right     | '0'                | 01                                     |
| TP length  | integer | 8              | right     | '0'                | variable                               |
| <b>Data Packet (DP)</b>                                  |         |                |           |                    |  |
| Telegram Type  | ascii   | 8              | right     | **'                | **TAGPOS                               |
| DP-Version Number  | integer | 2              | right     | '0'                | 02                                     |
| Number of positions                                      | hex     | 4              | right     | '0'                | 000A                                   |
| The next fields are repeated 'Number of positions' times |         |                |           |                    |  |
| Tag-ID   | hex     | 8              | right     | '0'                | 6400011F                               |
| X coordinate   | hex     | 8              | right     | '0'                | 00002710<br>Value in cm (here 10000cm) |
| Y coordinate   | hex     | 8              | right     | '0'                | Value in cm                            |
| Z coordinate   | hex     | 8              | right     | '0'                | Value in cm                            |
| Yaw (in 1/10 deg)  | hex     | 3              | right     | '0'                | 708 (=180 deg)                         |
| Pitch (in 1/10 deg)                                      | hex     | 3              | right     | '0'                | 708 (=180 deg)                         |
| Roll (in 1/10 deg)                                       | hex     | 3              | right     | '0'                | 000 (=0 deg)                           |
| GDOP Value in cm   | hex     | 4              | right     | '0'                | 007D (=1.25m)                          |
| Standard-Error Value in cm                               | hex     | 4              | right     | '0'                | 007D (=1.25m)                          |
| StdErr Valid?<br>0=false, 1=true                         | integer | 1              | right     | none               | 1                                      |

**Example**

## Telegram Type

TH0100000026\*\*TAGPOS020000

TH0100000076\*\*TAGPOS02000164000001000027100000271000000001708000000007D007D1

## TAGPOS Version 03

| Description  | Type    | Length (bytes) | Alignment | Padding characters | Example    |
|--|---------|----------------|-----------|--------------------|------------|
| <b>Transport Packet (TP)</b>                             |         |                |           |                    |            |
| Transport Header   | ascii   | 2              | fixed     | none               | TH         |
| TP-Version Number  | integer | 2              | right     | '0'                | 01         |
| TP length  | integer | 8              | right     | '0'                | variable   |
| <b>Data Packet (DP)</b>                                  |         |                |           |                    |            |
| Telegram Type  | ascii   | 8              | right     | '*'                | **TAGPOS   |
| DP-Version Number  | integer | 2              | right     | '0'                | 03         |
| Number of positions                                      | hex     | 4              | right     | '0'                | 000A       |
| The next fields are repeated 'Number of positions' times |         |                |           |                    |            |
| Tag-ID   | hex     | 8              | right     | '0'                | 6400011F   |
| X coordinate Value in cm                                 | integer | 16             | right     | '*'                | *****29473 |
| Y coordinate Value in cm                                 | integer | 16             | right     | '*'                | *****7004  |
| Z coordinate Value in cm                                 | integer | 16             | right     | '*'                | *****-300  |
| Yaw (in 1/10 deg)  | integer | 4              | right     | '*'                | 1800       |
| Pitch (in 1/10 deg)                                      | integer | 4              | right     | '*'                | *900       |
| Roll (in 1/10 deg)                                       | integer | 4              | right     | '*'                | ***0       |
| GDOP Value in cm   | integer | 8              | right     | '*'                | ***12500   |
| Standard-Error Value in cm                               | integer | 8              | right     | '*'                | 20         |
| StdErr Valid?<br>0=false, 1=true                         | integer | 1              | right     | none               | 1          |

**Example**

## Telegram Type

```
TH0100000026**TAGPOS030000
TH0100000111**TAGPOS03000164000001*****29473*****7004*****-
3001800*900**0***12500*****201
```

## TAGPOS Version 04

| Description  | Type    | Length (bytes) | Alignment | Padding characters | Example          |
|--|---------|----------------|-----------|--------------------|------------------|
| <b>Transport Packet (TP)</b>                             |         |                |           |                    |                  |
| Transport Header   | ascii   | 2              | fixed     | none               | TH               |
| TP-Version Number  | integer | 2              | right     | '0'                | 01               |
| TP length  | integer | 8              | right     | '0'                | variable         |
| <b>Data Packet (DP)</b>                                  |         |                |           |                    |                  |
| Telegram Type  | ascii   | 8              | right     | '**'               | **TAGPOS         |
| DP-Version Number  | integer | 2              | right     | '0'                | 04               |
| Number of positions                                      | hex     | 4              | right     | '0'                | 000A             |
| The next fields are repeated 'Number of positions' times |         |                |           |                    |                  |
| Tag-ID   | hex     | 16             | right     | '0'                | 0011CE006400011F |
| X coordinate Value in cm                                 | integer | 16             | right     | '**'               | *****29473       |
| Y coordinate Value in cm                                 | integer | 16             | right     | '**'               | *****7004        |
| Z coordinate Value in cm                                 | integer | 16             | right     | '**'               | *****-300        |
| Yaw (in 1/10 deg)  | integer | 4              | right     | '**'               | 1800             |
| Pitch (in 1/10 deg)                                      | integer | 4              | right     | '**'               | *900             |



| Description                      | Type    | Length (bytes) | Alignment | Padding characters | Example  |
|----------------------------------|---------|----------------|-----------|--------------------|----------|
| Roll (in 1/10 deg)               | integer | 4              | right     | '*'                | ***0     |
| GDOP Value in cm                 | integer | 8              | right     | '*'                | ***12500 |
| Standard-Error Value in cm       | integer | 8              | right     | '*'                | 20       |
| StdErr Valid?<br>0=false, 1=true | integer | 1              | right     | none               | 1        |

### Example

TH0100000026\*\*TAGPOS040000

TH0100000119\*\*TAGPOS0400010011CE0064000001\*\*\*\*\*29473\*\*\*\*\*700  
4\*\*\*\*\*-3001800\*900\*\*\*0\*\*\*12500\*\*\*\*\*201

Telegram Type

## TAGSTAT

Tag Battery Status Telegram.

This telegram is used within the ACSSupport package for transferring live tag battery status data from a production site to a test site (Location Transfer Service).

The location transfer service of the production site continuously sends TAGSTAT telegrams for single tags whenever their status changes.

Sender: ACS

Recipient: External system (Server MES, PLC...)

Acknowledgment: no

## TAGSTAT Version 01

| Description  | Type    | Length (bytes) | Alignment | Padding characters | Example  |
|--|---------|----------------|-----------|--------------------|----------|
| <b>Transport Packet (TP)</b>   |         |                |           |                    |          |
| Transport Header   | ascii   | 2              | fixed     | none               | TH       |
| TP-Version Number  | integer | 2              | right     | '0'                | 01       |
| TP length  | integer | 8              | right     | '0'                | 0000035  |
| <b>Data Packet (DP)</b>  |         |                |           |                    |          |
| Telegram Type  | ascii   | 8              | right     | '*'                | *TAGSTAT |
| DP-Version Number  | integer | 2              | right     | '0'                | 01       |
| Number of tags   | hex     | 4              | right     | '0'                | 0001     |
| The next fields are repeated 'Number of tags' times                                    |         |                |           |                    |          |
| Tag-ID   | hex     | 8              | right     | '0'                | 6400011F |
| Battery Status<br>One lower-case letter:<br>'o' = OK<br>'w' = Warning<br>'f' = Failing | ascii   | 1              | fixed     | none               | o        |

**Example**

```
TH0100000035*TAGSTAT0100016400011Fo
TH0100000053*TAGSTAT01000300000001o00000002w00000003f
```

## Telegram Type

### TAGSTAT Version 02

Tag-ID length changed, compared to V01 telegram

| Description  | Type    | Length (bytes) | Alignment | Padding characters | Example          |
|--|---------|----------------|-----------|--------------------|------------------|
| <b>Transport Packet (TP)</b>   |         |                |           |                    |                  |
| Transport Header   | ascii   | 2              | fixed     | none               | TH               |
| TP-Version Number  | integer | 2              | right     | '0'                | 01               |
| TP length  | integer | 8              | right     | '0'                | 0000043          |
| <b>Data Packet (DP)</b>  |         |                |           |                    |                  |
| Telegram Type  | ascii   | 8              | right     | '*'                | *TAGSTAT         |
| DP-Version Number  | integer | 2              | right     | '0'                | 02               |
| Number of tags   | hex     | 4              | right     | '0'                | 0001             |
| The next fields are repeated 'Number of tags' times                                    |         |                |           |                    |                  |
| Tag-ID   | hex     | 16             | right     | '0'                | 0011CE006400011F |
| Battery Status<br>One lower-case letter:<br>'o' = OK<br>'w' = Warning<br>'f' = Failing | ascii   | 1              | fixed     | none               | o                |

### Example

```
TH0100000043*TAGSTAT0200010011CE006400011Fo
TH0100000061*TAGSTAT0200030011CE0000000001o00000002w00000003f
```

## TAGSTRQ

Request to send the battery status for all known tags.

This telegram is used within the ACSSupport package by a test site to request live tag battery status data from a production site (Location Transfer Service).

The TAGSTRQ telegram can be sent by the test site requesting the status of all tags when the connection is established. This may be used for an initial synchronization. Optionally, a list of tag IDs may be given. The production site replies with a TAGSTAT telegram containing the battery status for the specified tags, or for all known tags if no tags were specified.

Sender: ACS or External system (Server MES or PLC...)

Recipient: ACS or External system (PLC, Station PC, Tool controller)

Acknowledgment: yes

### TAGSTRQ Version 01

| Description                                       | Type    | Length (bytes) | Alignment | Padding characters | Example  |
|---|---------|----------------|-----------|--------------------|----------|
| <b>Transport Packet (TP)</b>                      |         |                |           |                    |          |
| Transport Header                                  | ascii   | 2              | fixed     | none               | TH       |
| TP-Version Number                                 | integer | 2              | right     | '0'                | 01       |
| TP length   | integer | 8              | right     | '0'                | 00000034 |
| <b>Data Packet (DP)</b>                           |         |                |           |                    |          |
| Telegram Type                                     | ascii   | 8              | right     | '*'                | *TAGSTRQ |
| DP-Version Number                                 | integer | 2              | right     | '0'                | 01       |
| Number of tags                                    | hex     | 4              | right     | '0'                | 0001     |
| The next field is repeated 'Number of tags' times |         |                |           |                    |          |
| Tag-ID  | hex     | 8              | right     | '0'                | 6400011F |

### Example

```
TH0100000026*TAGSTRQ010000
TH0100000034*TAGSTRQ0100016400011F
```

## Telegram Type

### TAGSTRQ Version 02

Tag-ID length changed, compared to V01 telegram

| Description                                       | Type    | Length (bytes) | Alignment | Padding characters | Example          |
|---|---------|----------------|-----------|--------------------|------------------|
| <b>Transport Packet (TP)</b>                      |         |                |           |                    |                  |
| Transport Header                                  | ascii   | 2              | fixed     | none               | TH               |
| TP-Version Number                                 | integer | 2              | right     | '0'                | 01               |
| TP length   | integer | 8              | right     | '0'                | 00000042         |
| <b>Data Packet (DP)</b>                           |         |                |           |                    |                  |
| Telegram Type                                     | ascii   | 8              | right     | **                 | *TAGSTRQ         |
| DP-Version Number                                 | integer | 2              | right     | '0'                | 02               |
| Number of tags                                    | hex     | 4              | right     | '0'                | 0001             |
| The next field is repeated 'Number of tags' times |         |                |           |                    |                  |
| Tag-ID  | hex     | 16             | right     | '0'                | 0011CE006400011F |

### Example

```
TH0100000026*TAGSTRQ020000  
TH0100000042*TAGSTRQ0200010011CE006400011F
```

## APPCONF

Application configuration telegram

This telegram can be used by external systems to provide dynamic configuration information. It is up to the recipient to implement handlers for the configuration change. See the table below for the configuration changes ACS will handle.

Sender: ACS or External system (Server MES or PLC...)

Recipient: ACS or External system (PLC, Station PC, Tool controller)

Acknowledgment: yes

| Description   | Type    | Length (bytes) | Alignment | Padding characters | Example                                      |
|---|---------|----------------|-----------|--------------------|--|
| <b>Transport Packet (TP)</b>  |         |                |           |                    |  |
| Transport Header  | ascii   | 2              | fixed     | none               | TH   |
| TP-Version Number   | integer | 2              | right     | '0'                | 01   |
| TP length   | integer | 8              | right     | '0'                | variable                                     |
| <b>Data Packet (DP)</b>   |         |                |           |                    |  |
| Telegram Type   | ascii   | 8              | right     | **                 | *APPCONF                                     |
| DP-Version Number   | integer | 2              | right     | '0'                | 01   |
| Configuration Group information   | ascii   | 32             | right     | **                 | Value from an agreed list, e.g. AssemblyLine |
| Configuration Object information  | ascii   | 32             | right     | **                 | Object identifier, e.g. Line45               |
| Number of configuration attributes                                      | integer | 4              | right     | '0'                | 0002   |
| The next fields are repeated 'Number of configuration attributes' times |         |                |           |                    |  |
| Attribute name  | ascii   | 32             | right     | **                 | Value from an agreed list, e.g. LinelsMoving |
| Attribute value length  | integer | 4              | right     | '0'                | 0004   |
| Attribute value   | ascii   | variable       | right     | none               | true   |

### Example

## Telegram Type

```
TH0100000169*APPCONF01*****AssemblyLine*****Line45
0002*****LineSpeed0003600*****LineIsMoving0004true
```

## Configurations handled by ACS

| Configuration Group | Attribute Name  | Attribute type and domain       | Description   |
|---------------------|-----------------|---------------------------------|---|
| AssemblyLine        | LineIsMoving    | Boolean, as [0, 1, false, true] | For assembly lines configured with 'Dynamic Configuration Required' = yes, set the line as moving (1, true) or non moving (0, false)  |
| AssemblyLine        | LineSpeed       | Integer, e.g. 600 [cm/min]      | For assembly lines configured with 'Dynamic Configuration Required' = yes, set the line to the configured value   |
| AssemblyLine        | MaintenanceMode | Boolean, as [0, 1, false, true] | Set the maintenance mode for an assembly line (1, true) or remove maintenance mode (0, false). An assembly line in maintenance mode will ignore tag updates and not move products   |
| AssociationZone     | MaintenanceMode | Boolean, as [0, 1, false, true] | Set the maintenance mode for an association zone (1, true) or remove maintenance mode (0, false). An association zone in maintenance mode will ignore tag updates and not perform product-tag associations                  |
| DisassociationZone  | MaintenanceMode | Boolean, as [0, 1, false, true] | Set the maintenance mode for a disassociation zone (1, true) or remove maintenance mode (0, false). A disassociation zone in maintenance mode will ignore tag updates and not perform any product or tag related activities |