

Standard Equipment Interface

Revision history

Version	Date	Author	Description
2.0	14-11-2023	SML	Initial version (merge from other variants of the interface specification).
2.1	06-02-2024	AEF	Add info-request for Locations. Fixed some issues in example telegrams.
2.2	12-06-2024	AEF	Added resource ID to error telegrams.
2.3	18-08-2024	MIV	Added common types for locations & changed telegrams.
2.4	28-02-2025	MIV	Major changes to interfaces and document structure
	24-06-2025	PDA	Release

Review history

Version	Date	Reviewer	Notes / description
2.0	16-11-2023	AEF	Review
2.1	16-02-2024	SML	Review
2.2	13-06-2024	MIV	Review
2.3	19-08-2024	AEF	Review
2.4	12-03-2025	AEF	Review
	24-06-2025	PDA	Release

Table of Contents

1 Introduction	4
1.1 References	4
1.2 Glossary	4
1.3 Copyright and right to use under Apache License 2.0	4
1.4 Background / 5G-Robot	4
2 Telegram Format	6
2.1 Property Name Formatting	6
2.2 List of Relevant Error Codes	6
2.3 Common Properties	7
2.4 Common Data Types	7
2.4.1 Item ID	7
2.4.2 Location ID	7
2.4.3 Location Type	7
2.4.4 Location State	8
2.4.5 Location Config	8
2.4.6 Location Information	9
2.4.7 Container Content	10
2.4.8 Container	11
2.4.9 Capability Property	12
3 Common Telegrams	14
3.1 Info-request Telegram (HLC <-> Equipment)	14
3.2 System State Handling	15
3.2.1 System State (Equipment -> HLC)	15
3.2.2 Request State Change (HLC -> Equipment)	16
3.2.3 Alarm State (Equipment -> HLC)	16
3.2.4 Active Alarms (Equipment -> HLC)	17
3.2.5 Locations (Equipment -> HLC)	17
3.2.6 Capabilities (Equipment -> HLC)	18
3.2.7 Update Capability (Equipment <-> HLC)	20
3.3 Item/Container Updates and Management	21
3.3.1 Item/Container at Location (HLC <-> Cell)	21
3.3.2 Container Ready for Pickup (Cell <-> HLC)	22
3.3.3 Container Picked Up (HLC <-> Cell)	23
3.4 Resource Access	24
3.4.1 Resource Access Request (HLC <-> Cell)	24
3.4.2 Resource Access Reply (HLC <-> Cell)	25
3.4.3 Resource Access Complete (HLC <-> Cell)	26
4 Scenarios and Example Flows	27
4.1 Resource Access Example	27

1 Introduction

This document describes the communication protocol, telegram formatting and defines a set of common telegrams that is used to interface between the HLC and equipment. Specialized equipment interfaces for e.g. production cells and mobile robot top-modules can extend this document with domain specific telegrams, examples, and appendices. It is an extension of the Standard Interfaces [1] document.

1.1 References

ID	Document	Description
[1]	Standard Interfaces v1.0	The base interface description used for all the open interfaces. Found on Intelligent Systems website: https://www.intelligentsystems.dk/products-keep-customers-at-forefront-of-technology/
[2]	What is PackML?	https://www.omac.org/packml
[3]	GS1 Global Returnable Asset Identifier (GRAI)	https://www.gs1.org/standards/id-keys/grai

1.2 Glossary

Abbreviation	Description
Equipment	Examples of specializations of equipment are production and palletizing cells, and top-modules.
HLC	H igh- L evel C ontrol system - e.g. WCS/WMS, MES or MFS.

1.3 Copyright and right to use under Apache License 2.0

The copyright on this document and any contained specifications, designs, etc. belongs to the company: Intelligent Systems A/S, Havnevej 11, DK-9560 Hadsund, DENMARK.

Any referenced 3rd party designs, technologies and or other IP shall remain with the original owners.

The reference architecture, designs and the included open standard integration interfaces are open and free to use under Apache License 2.0. For details please see <https://www.apache.org/licenses/LICENSE-2.0>.

1.4 Background / 5G-Robot



Parts of this document / release was made in the **5G-Robot** project also known under the long name **5G-ENABLED AUTONOMOUS MOBILE ROBOTIC SYSTEMS** - the largest innovation project that has been launched under the Innovation Fund Denmark's (IFD) Grand Solutions program.

The groundbreaking project united Denmark's leading robot, automation and factory digitalization companies as technology vendors, research partners and industry-leading end-user companies.



Illustration: Project partner logos.

The aim of the project was to revolutionize manufacturing - paving the way to smart production and smart factories and the application of a number of new technologies in production and manufacturing including 5G wireless communication, cloud and edge computing and digital twin.

Intelligent Systems played a leading role in the project, providing the glue that ties the robotic solutions of the partners together making the work as one - i.e. one connected integrated intelligent manufacturing system.

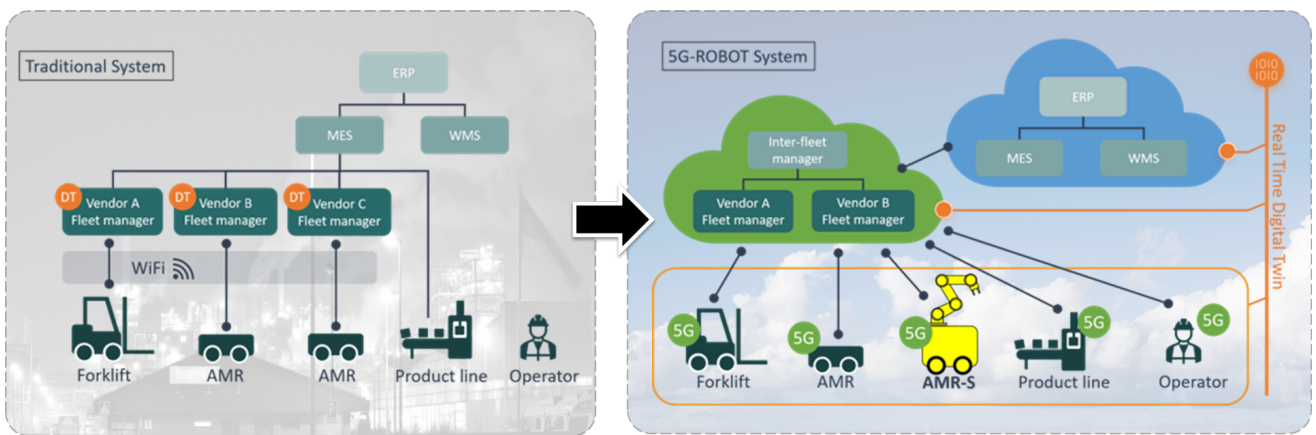


Illustration: The aim is to revolutionize manufacturing paving the way to smart production and smart factories.

Read more about the 5G-Robot project here: <https://www.5gsmartproduction.aau.dk/5g-robot>

2 Telegram Format

This section describes the structure and format of the message telegrams.

2.1 Property Name Formatting

All interfaces that derive from this specification must replace hyphens (-) with underscores (_). This is to allow for easier integration to PLCs which rely on libraries that map the property names to variables. With hyphens being unsupported, this will lead to several issues during implementation.

The same change applies to telegrams in the Standard Interface document [1].

2.2 List of Relevant Error Codes

The following list of error codes are referenced in this interface. The relevant Error ID's will be listed for each telegram of which it is relevant.

Error ID	Error Message	Reason
INVALID_MESSAGE	Invalid Message	The request could not be parsed.
INVALID_STATE	Invalid State	Equipment is not in a state to execute the request. Example: Cannot Reset if the equipment is in an Idle state.
LOCATION_NOT_FOUND	Location not found	The <i>location_id</i> was not recognized.
MESSAGE_TOO_LARGE	Message too large	Message length of request exceeds the maximum supported size on the device.
NOT_FOUND	Item Not Found	The <i>resource_id</i> was not recognized.
UNKNOWN_REQUEST	Unknown Request	The <i>request_id</i> was not recognized.
INVALID_VALUE	Invalid value	The given value is outside the bounds possible.
NO_ERROR	No error	Telegram was accepted without any issues.
PARAM_CHANGE_NOT_ALLOWED	Parameter change not allowed	A given parameter can only be changed on or by the equipment itself.
REPEATED_TELEGRAM	Sequence number same as the last one	Sent, when a telegram is – based on sequence number - considered “repeated”. Telegram will <u>not</u> be processed.
SEQ_TOO_HIGH	Sequence number too high	Sent when a sequence number is larger than current value is received. The telegram will be processed and the sequence counter will be adjusted by the receiver.
SEQ_TOO_LOW	Sequence number too low	Sent when the sequence is too small. Telegram is considered too low when the sequence number is less than the expected sequence number. Else it is considered as a repeat telegram. Telegram will <u>not</u> be processed.
UNKNOWN_PARAM	Unknown parameter	The given parameter does not exist in the equipment.

2.3 Common Properties

The following list contains commonly used property types in telegrams.

Property name	Type	Description
item_id	ItemID	<p>An ID for the item tracked by the equipment. If the equipment cannot determine the ID for any reason, a null value can be used.</p> <p>The ID is determined by the equipment when the item enters the system/equipment, and must be unique as long as it is handled and tracked by the equipment. That is, an ID is assigned once it is first tracked until it exits the equipment, e.g. by being transported or palletized.</p>
location_id	LocationID	<p>An ID to reference a specific location.</p> <p>The ID for a given location should be known by both the HLC and the equipment. This configuration is not part of the interface specification.</p>

2.4 Common Data Types

2.4.1 Item ID

Data type: **ItemID**

A string representing a unique identification of an item. This can be provided by the HLC through e.g. serial numbers or another value that is unique for the item. Otherwise, this can be provided by the cell (e.g. through an incrementing value).

2.4.2 Location ID

Data type: **LocationID**

A string representing a unique identification of a location. This ID can be defined by either side - depending on who creates the locations. The Location ID **MUST** be unique for all locations handled on any given communications channel. The HLC will be able to use both the identifier of the communications channel and the Location ID to identify the specific location.

So for any given channel all locations must be uniquely identifiable. So if a system has 3 automated storages and each of them has a location called 1-1-1, then this is uniquely identifiable. If on the other hand all 3 automated storages communicate on a single communications channel, then the Location ID must have more information to be uniquely identifiable - for example by prefixing a number uniquely identifying the individual automated storages.

2.4.3 Location Type

Data type: **LocationType**

Location type is a string used to define how a given location is to be used.

The following values can be used:

Location Type	Description
storage-location	Location used to store containers.
infeed	Location where containers going into storage is placed.
outfeed	Location where containers coming out from storage is placed.
infeed-outfeed	Location that can be used either as an infeed or an outfeed.

2.4.4 Location State

Data type: **LocationState**

This describes the state of a location or area, so that the equipment can utilize it accordingly. This also means that all states do not have to be utilized if not relevant.

The following values can be used:

Location State	Description
empty	The location is empty.
ongoing	The location has containers, but are not allowed to be handled. For example, if the equipment is currently operating in the area, this status is used to tell the HLC that it cannot reserve this area.
ready	The containers at the location are ready to be handled and that the cell or HLC (through external equipment) may operate there. It also means that there will be no changes to the location/area until the equipment has handled the container(s). It is NOT allowed to make changes to the location/area at this point, even though there might still be room for an additional container.
error	There is an issue with the location and the equipment is unable to clear it.
occupied	The location (or area) is unable to handle any more containers (i.e. it is full).
blocked	The location (or area) is marked as being unavailable by e.g. an operator.
unknown	The location is known by the equipment, but it does not know if there are any containers on them.

2.4.5 Location Config

Data type: **LocationConfig**

This information is used to create or adjust locations in the HLC.

Properties

Property	Type	Description
location_type	LocationType	The type of location. See LocationType .
external_id	String or null	If location is controlled by an external system, its name is given here so it can be linked in the HLC. If it is not controlled by an external system, the value is null.

The coordinates given are to be used for future reference to the location by the HLC.

Examples

```
{
  "location_type": "storage"
}
```

```
{
  "location_type": "infeed",
  "external_id": "7025.1"
}
```

2.4.6 Location Information

Data type: **LocationInformation**

This information is used to send information about locations to the HLC.

Properties

Property	Type	Description
id	String	ID of the location. The ID may also be a name of an area and thereby be able to group locations by area definitions.
item_id	ItemID or null	ID of the item at the location, if present. If a container is placed on this location (and the GRAI is included), or if nothing is present at this location, this value is null or omitted.
grai	String or null	GRAI [3] of the container at the location, if present. If an item is placed directly on the location (and an item_id is included), or if nothing is present at the location, this value is null or omitted. Below is a list of specialized codes used as GRAI but not GRAI.
contents	Object (String, String) or null	Included if GRAI is present. Otherwise it is null, an empty array or omitted. Key-value pair describing each item in the carrier.

		The key/property is an id relevant to the placement in the carrier, this can e.g. be a simple numerical index for each item. The value is an id for the item in the carrier. The id can be a serial number for the item or another identifier to inform the HLC what item it is.
state	LocationState or null	See LocationState . State of a location to indicate its readiness.
location_config	LocationConfig or null	See LocationConfig . Definition of a given location sent to the HLC. This is used to create or update the location definition in the HLC if possible.
locations	List of LocationInformation, or null	Locations may be defined hierarchically and this list makes it possible. If not applicable, the value is null.

Specialized GRAI codes:

Code	Description
noread	Used to indicate that the specified container on the given location was a no-read at the last scanner. If the equipment does not track containers the erroneous container is still guided towards an error handling station.
unknown_container	If a container is found on the Equipment at a place that does not have a scanner, it can be reported to the HLC as an Unknown container. This lets the HLC guide it towards an error handling station. If the container is scanned at a later point the HLC will redirect it accordingly.

The HLC will not create these containers specifically since they cannot be identified uniquely. These codes simply tell the HLC what to do at a given point. The same container may be scanned correctly later on and the container will then be redirected accordingly.

2.4.7 Container Content

Data type: **ContainerContent**

This data type contains information about the content of a container.

Properties

Property	Type	Description
item_id	ItemID or null	Optional. ID of the item.
quantity	Integer or null	Optional. Quantity of the item. The unit of measure is pieces (PCS).

properties	Object String, String	Key-value pair describing something about each item in the container. This can, for example, be used for specifying where in a container an item is located. The usage is subject to definition in specific implementations.
------------	--------------------------	--

Examples

```
{
  "item_id": "10243-001",
  "quantity": "1",
  "properties": {
    "place": "1"
  }
}
```

2.4.8 Container

Data type: **Container**

This has information about containers and their content.

Properties

Property	Type	Description
grai	String	GRAI of the carrier(s) [3] .
container_type	String	Type of container (box, tray etc.).
sub_location	String or null	Optional. If a location can have multiple serviceable locations (for example, a mobile robot with multiple locations), this property can be used to specify where on the location the container is located. If not required, i.e. there is no ambiguity for the location, this value is null.
qa_pass	Boolean or null	Optional. Only applicable if the cell is capable of performing quality assurance checks. If not, this value will always be null or omitted.
contents	List of ContainerContent	Specifies what is currently placed on the container.

Examples

```
{
  "grai": "01234567890123456789",
  "container_type": "TRAY",
  "contents": [
```

```
{
  "item_id": "10243-001",
  "quantity": "1",
  "properties": {
    "place": "1"
  }
},
{
  "item_id": "10243-004",
  "quantity": "1",
  "properties": {
    "place": "2"
  }
}
]
```

```
{
  "grai": "01234567890123456789",
  "container_type": "TRAY",
  "sub_location": "left",
  "contents": [
    {
      "item_id": "99955-001",
      "quantity": "1",
      "properties": {}
    }
  ]
}
```

```
{
  "grai": "01234567890121212121",
  "container_type": "TRAY",
  "qa_pass": true,
  "contents": [
    {
      "item_id": "10243-001",
      "quantity": "1",
      "properties": {
        "place": "1"
      }
    }
  ]
}
```

2.4.9 Capability Property

Data type: **CapabilityProperty**

The property is a key-value pair of strings. This ensures that the telegram is generic but still able to report specific values usable by the equipment in question. All capabilities must be specified in detail by the vendor so that relevant values may be used by the HLC to implement relevant functionality using the capabilities.

Property	Type	Description
name	String	Name of the property.
value	String	Value of the property. Non-string values should be sent as the string-representation of that type.
type	String	The type of the value, in lowercase. Must be one of the formats described in section 3.1. Used by the receiver to validate and parse the value correctly.

Example

The following example shows properties of various data types.

```

{
  "name": "type",
  "value": "Conveyor",
  "type": "string"
}

{
  "name": "mat_group_1_speed",
  "value": "5",
  "type": "number"
}

{
  "name": "current_spee",
  "value": "2.4",
  "type": "number"
}

{
  "name": "is_active",
  "value": "true",
  "type": "boolean"
}

{
  "name": "startup_time",
  "value": "2023-01-19T09:01:50Z",
  "type": "date"
}

{
  "name": "request_id",
  "value": "62d7a39a-97e2-11ed-a8fc-0242ac120002",
  "type": "uuid"
}

{
  "name": "max_speed",
  "value": "null",
  "type": "null"
}

```

3 Common Telegrams

3.1 Info-request Telegram (HLC <-> Equipment)

Telegram Type ID: info-requests

Telegram transmitted by the HLC or equipment. The receiver should respond with a corresponding telegram on the relevant topic.

These messages are primarily used for (re-)synchronization. They will not be used for e.g. requesting a state change for the equipment. Note, additional request types can be defined for specialized equipment that extends from this interface specification.

Request Type	Value	Sender	Response telegram	Description
System State	state	HLC	System State	Request the current state of the equipment.
Active Alarms	active-alarms	HLC	Active Alarms	Request status of current active alarms in the equipment. If the status of a specific alarm is desired, then the optional parameter can be used to specify the alarm ID.
Locations	locations	HLC	Locations	Request information from the equipment about controlled locations. If the information of a specific location or locations in a specific area is desired, then the optional parameter can be used to specify the location or area ID.
Capabilities	capabilities	HLC	System Capabilities	Request list of capabilities of the equipment.

Properties

Property	Type	Description
request	String	Requested resource. Can be one of the values described above.
parameter	String or null	Optional. Parameter used in the request to specify a resource.

Example

```
Without a parameter
{
  "header": {...},
  "request": "state"
}
```

```

With a parameter
{
  "header": {...},
  "request": "state",
  "parameter": "1"
}

```

3.2 System State Handling

3.2.1 System State (Equipment -> HLC)

Telegram Type ID: system.state

Sent by the equipment when state has changed or when requested by the HLC using the [Info-request telegram](#).

The states follow the PackML state model [2] in addition to an "Offline" state. The offline state should be used just before the equipment or HLC is turned off.

When the equipment connects to the broker, it should send an initial status message.

Properties

Property	Type	Description
state	String	Current state of the equipment. Follows the PackML state model [1].
parameter	String or null	Optional. Parameter that the equipment may use to add extra information about the state. The property is by default not used, and is specified in the interface specializations if relevant.

Example

```

Equipment with active operation (operation identified in parameter)
{
  "header": {...},
  "state": "Execute",
  "parameter": "285"
}

```

3.2.2 Request State Change (HLC -> Equipment)

Telegram Type ID: system.state.request-change

Request from the HLC for the equipment to change its state. The equipment should only accept the request when it is a state to do so (see PackML [1]). Some equipment may only use parts of the states of PackML and other equipment may have extra states that are not part of PackML. This is explained in the individual interfaces.

After receiving the request, the equipment should send a [System State](#) telegram with its new state.

Properties

Property	Type	Description
state_change	String	State change command. Follows the PackML standard.

Errors

The following error types can be returned by the HLC for this request:

- ERR_INVALID_STATE
- ERR_MESSAGE_TOO_LARGE
- ERR_INVALID_MESSAGE

Example

```
{
  "header": {...},
  "state_change": "Suspend"
}
```

3.2.3 Alarm State (Equipment -> HLC)

Telegram Type ID: system.alarm

Telegram transmitted by the equipment when the alarm status alters.

The alarms are configured on the equipment and HLC and are not part of this specification.

Properties

Property	Type	Description
alarm_id	String	ID of the alarm.
active	Boolean	Indicates if the alarm is active (true) or inactive (false).

Example

```
{
  "header": {...},
  "alarm_id": "50",
  "active": true
}
```

3.2.4 Active Alarms (Equipment -> HLC)

Telegram Type ID: system.active-alarms

Telegram transmitted by the equipment when requested using a [Info-request telegram](#). This is typically used at start-up or after connection is reestablished after a connection loss to ensure that HLC is updated on current alarm states.

The telegram contains an array of all active alarm IDs. If there are no active alarms, an empty list is provided. The HLC will deactivate any active alarms for equipment not in the received list.

The alarms are configured on the equipment and HLC and are not part of this specification.

Properties

Property	Type	Description
alarms	List of String	Array of alarms IDs.

Example

```
{
  "header": {...},
  "alarms": [
    "50", "62"
  ]
}
```

3.2.5 Locations (Equipment -> HLC)

Telegram Type ID: system.locations

Telegram transmitted by the equipment when requested using a [Info-request telegram](#), or when location information updated. This is typically used at start-up or after connection is reestablished after a connection loss to ensure that HLC has the correct information for the locations on the equipment and which items/containers are present.

The telegram contains an array of all locations for the equipment and items/containers on them, if any.

The locations are configured on the equipment and HLC and are not part of this specification.

Properties

Property	Type	Description
locations	List of LocationInformation	Array of locations.

Example

Note that comments have been added to the example to describe each possible combination.

```

{
  "header": {...},
  "locations": [
    (Example with an Item directly on a location)
    {
      "id": "1",
      "item_id": "ABC-123",
      "grai": null,
      "contents": null
    },
    (Example with a container on the location, with contents)
    {
      "id": "2",
      "item_id": null,
      "grai": "01234567890066",
      "contents": {
        "1": "ABC-123",
        "2": "ABC-456"
      }
    },
    (Example for a location with no items or containers on it)
    {
      "id": "3",
      "item_id": null,
      "grai": null,
      "contents": null
    }
  ]
}

```

3.2.6 Capabilities (Equipment -> HLC)

Telegram Type ID: system.capabilities

Sent by the equipment when requested by the HLC. The equipment profile should refer to the relevant Equipment Specification domain entities so that the HLC can properly delegate operations to equipment which it can execute. It also provides general information about what features the equipment is capable of handling.

With exception of the equipment type and equipment version, all values are sent as key-value pairs.

Properties

Property	Type	Description
equipment_type	String	Main part of the equipment type that identifies product name and producer if necessary.
equipment_version	String	The version of the equipment. This can be anything from software version, model number, With this information it is possible to determine exactly what can be sent or received from the equipment.
capabilities	List of CapabilityProperty	See CapabilityProperty .

Example

```
{
  "header": {...},
  "equipment_type": "Pallet lift",
  "equipment_version": "Sw 3.0.187",
  "capabilities": [
    {
      "name": "Mat_group_1_speed",
      "value": "5",
      "type": "number"
    },
    {
      "name": "Mat_group_1_accel",
      "value": "8",
      "type": "number"
    },
    {
      "name": "Mat_group_2_speed",
      "value": "3",
      "type": "number"
    },
    {
      "name": "Mat_group_2_accel",
      "value": "4",
      "type": "number"
    },
    {
      "name": "Mat_group_3_speed",
      "value": "1",
      "type": "number"
    },
    {
      "name": "Mat_group_3_accel",
      "value": "2",
      "type": "number"
    }
  ]
}
```

3.2.7 Update Capability (Equipment <-> HLC)

Telegram Type ID: system.capabilities.update

Telegram transmitted by either system when a change in a capability has been made. Changes may be made in both systems and then updated in the other system. In case a parameter may only be changed in one system, this has to be agreed upon in every single case.

Properties

Property	Type	Description
capability_properties	List of CapabilityProperty	See CapabilityProperty .

Errors

The following error types can be returned by the HLC for this request:

- INVALID_VALUE
- PARAM_CHANGE_NOT_ALLOWED
- UNKNOWN_PARAM

Example

```
{
  "header": {...},
  "capability_properties": [
    {
      "name": "Mat_group_1_accel",
      "value": "10",
      "type": "number"
    },
    {
      "name": "Mat_group_2_speed",
      "value": "4",
      "type": "number"
    },
    {
      "name": "Mat_group_2_accel",
      "value": "6",
      "type": "number"
    }
  ]
}
```

3.3 Item/Container Updates and Management

Management of items or containers is a common feature for all equipment.

All equipment can update an item or container location. Also reporting a container ready for pickup is possible from all equipment.

3.3.1 Item/Container at Location (HLC <-> Cell)

Telegram Type ID: item.location

Telegram transmitted by the production cell or the HLC when an item/container arrives at a location. This includes both locations that are only accessible by the production cell and locations that external equipment can interact with.

Sent when an item/container arrives at a location. This also includes when other messages relating to the item/container are sent, such as scans at an ID point.

If the item ID cannot be determined, a null value is used.

Properties

Property	Type	Description
location_id	String	ID of the location.
sub_location_id	String or null	Optional. If a location can have multiple serviceable locations (for example, a mobile robot with multiple locations), this property can be used to specify where on the location the carrier is located. If not required, i.e. there is no ambiguity for the location, this value is null.
item_id	ItemID or null	An ID for the item tracked by the cell.
scanned_id	String or null	Scanned container ID or null if read failed.

Example

```
Tracked item
{
  "header": {...},
  "location_id": "15",
  "sub_location_id": "1",
  "item_id": "12"
}
```

```
Untracked item
{
  "header": {...},
  "location_id": "15",
  "item_id": null
}
```

3.3.2 Container Ready for Pickup (Cell <-> HLC)

Telegram Type ID: item.pickup.available

Sent in one of the two following scenarios:

1. By the HLC when materials are ready for pickup by the cell.
2. By the cell when items are produced and are ready for pickup.

When picking up the container, the telegrams in [Accessing Resources on the Equipment](#) can be used. If coordination is not required when picking up the materials (e.g. if they are out of reach of the production cell), using these telegrams are not required.

After the container has been picked up, the [Container Picked Up](#) telegram is sent.

Properties

Property	Type	Description
location	String	ID of the location for picking up the container.
containers	List of Container	Containers ready for pickup.

Errors

The following error types can be returned by the HLC for this request:

- ERR_LOCATION_NOT_FOUND

Example

```
{
  "header": {...},
  "location": "7",
  "containers": [
    {
      "grai": "01234567890123456789",
      "container_type": "TRAY",
      "sub_location": "1",
      "contents": [
        {
          "item_id": "ABC-1567-001",
          "quantity": 1
        },
        {
          "item_id": "ABC-1567-002",
          "quantity": 1
        }
      ]
    }
  ]
},
{
  "grai": "01234567890123456799",
  "container_type": "TRAY",
  "sub-location": "2",
  "contents": [
```

```

    {
      "item_id": "ABC-1567-003",
      "quantity": 1
    }
  ]
}
]
}

```

3.3.3 Container Picked Up (HLC <-> Cell)

Telegram Type ID: item.pickup.completed

Sent by the HLC or cell when a container has been picked up.

This telegram should only be used after the [Container Ready for Pickup](#) telegram has been sent for the location.

To ensure the container can be picked up, the telegrams in Container Transport, or [Accessing Resources on the Equipment](#) in combination with [Item/Container at Location](#) can be used.

Properties

Property	Type	Description
location	String	ID of the location for picking up the container.
grai	List of String	GRAIs of the picked up carriers [3] .

Errors

The following error types can be returned by the HLC for this request:

- ERR_LOCATION_NOT_FOUND

Example

```

{
  "header": {...},
  "location": "7",
  "grai": [
    "01234567890123456789"
  ]
}

```

3.4 Resource Access

When access to a resource is controlled by an external source it is possible to request access to the resource. This is not limited to access from the HLC's side, it can also be an equipment that wishes access to another equipment resource.

Request for access goes through the HLC if relevant. This means that access from equipment A to equipment B's resources will go through the HLC. In this example equipment A sends ResourceAccessRequest to the HLC, which then sends a ResourceAccessRequest to equipment B.

That access request goes through the HLC ensures that resources on any equipment with interface to the HLC is available to any other equipment interfacing with the HLC - as well as the HLC itself also.

In some cases different equipment may interact with one another directly. This is agreed upon by the vendors directly and the HLC does not expect to know of these resources.

Resource Access may also be relevant to operators controlling equipment. If an operator needs access to the equipment itself or just parts of the equipment then this is possible using these resource access telegrams.

Requesting resource access is primarily to be used for access on stationary equipment. This means that mobile equipment that confirms placement at a given place has by default granted resource access to whatever resource it manages

3.4.1 Resource Access Request (HLC <-> Cell)

Telegram Type ID: resource.request

Telegram sent by the HLC or cell. Request to access a resource at a given ID, such as for example a location ID.

The owner of the resource, e.g. the cell if it is a buffer on the robot cell, can then accept or reject the request using the [Resource Access Reply](#).

If the HLC receives this and does not control the resource the HLC will then create its own ResourceAccessRequest to whichever equipment that controls the resource requested in the first place.

All new request messages for a resource should have a new request-id. Updates to a request (i.e. accepted, rejected, and completed) should use the same request id.

Properties

Property	Type	Description
request_id	UUID	ID of the request.
resource_id	String	ID of the resource Can be UUID.

Errors

The following error types can be returned by the HLC for this request:

- ERR_NOT_FOUND

Example

```
{
  "header": {...},
  "request_id": "62d7a39a-97e2-11ed-a8fc-0242ac120002",
  "resource_id": "59383fa0-6420-434a-91e4-cfffc9b311a9"
}
```

3.4.2 Resource Access Reply (HLC <-> Cell)

Telegram Type ID: resource.reply

Once access to a location has been requested with the [Resource Access Request](#), the owner of the resource is able to either accept or reject the request. **Note: if the request is accepted, the requester is allowed to access the resource immediately. Hence, the owner of the resource should be in a state that does not cause any collisions with the requester once the request is accepted.**

The reply should use the same request ID as the one provided in the original request.

When the requester no longer requires access to the resource, a [Resource Access Complete](#) telegram.

Properties

Property	Type	Description
request_id	UUID	ID of the request.
response	Boolean	True if accepted, false if rejected.

Errors

The following error types can be returned by the HLC for this request:

- ERR_UNKNOWN_REQUEST

Example

```
{
  "header": {...},
  "request_id": "62d7a39a-97e2-11ed-a8fc-0242ac120002",
  "response": true
}
```

3.4.3 Resource Access Complete (HLC <-> Cell)

Telegram Type ID: resource.completed

Once the requester has finished using the resource, this telegram should be sent by the requester to the owner of the resource. **Note: This telegram should be sent after the requester is no longer working in the area of the resource to ensure that no collision occurs with the owner of the resource.**

Furthermore, if containers/items have been transported to/from the resource, an Item/Container at Location telegram should also be sent to inform the owner of the change that was made to the resource.

The completion telegram should use the same request id as the one provided in the original request (See [Resource Access Request](#)).

Properties

Property	Type	Description
request_id	UUID	ID of the request.

Errors

The following error types can be returned by the HLC for this request:

- ERR_UNKNOWN_REQUEST

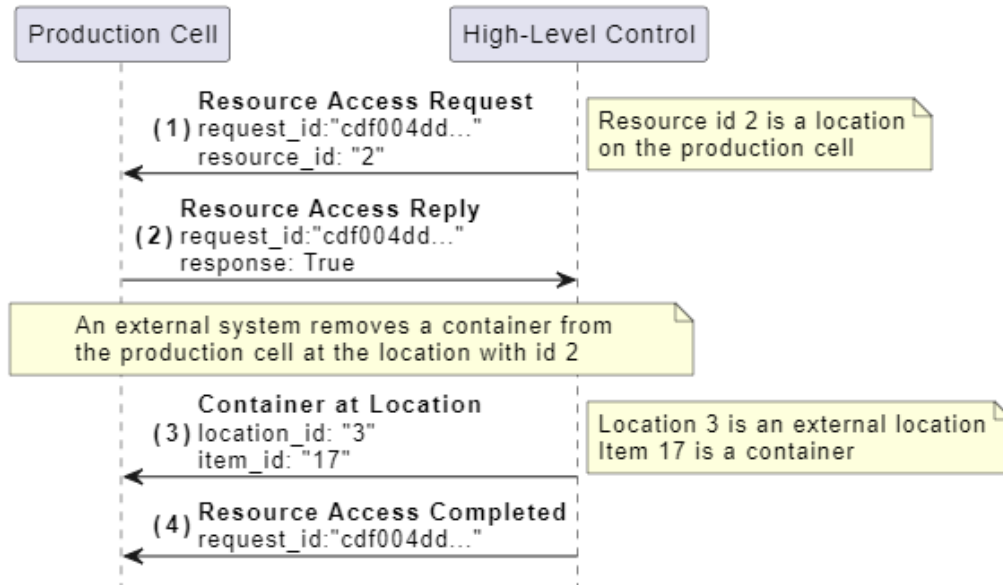
Example

```
{
  "header": {...},
  "request_id": "62d7a39a-97e2-11ed-a8fc-0242ac120002"
}
```

4 Scenarios and Example Flows

4.1 Resource Access Example

This scenario uses the telegrams in the Resource Access section to get access to and remove a container from an output buffer on a production cell. This request is required as the production cell operates in that area when producing a batch.



- (1) The HLC sends a request to access a resource with id "2". The request id is generated by the HLC, and is used whenever this specific resource request is to be referenced
- (2) The production cell clears from the resource (e.g. if it was in the way) and accepts the request
- (3) A container with id "17" has been moved from the output buffer location "2" to an external location "3". The item id is included to inform the cell that the item has exited the operating area of the cell.
- (4) The HLC informs the cell that it no longer requires access to the resource.