International Standard

ISO 23725

Autonomous system and fleet management system interoperability

Interopérabilité du système autonome et du système de gestion de la flotte

First edition
2024-08

Reference number
ISO 23725:2024(en)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 82, *Mining*, SC 8, *Advanced automated mining systems.*

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

A strategic objective for the global mining industry is to develop standards to support holistic, integrated and interoperable mine operating automation that will improve mining operations efficiency. Interoperability enables many benefits; optimal interaction of mine operating equipment and processes, integration of upstream process information (e.g. exploration, resource modelling and planning), minerals extraction and downstream processes (e.g. refining, smelting and transportation) to increase levels of operational efficiency. This document progresses interoperability in the global mining industry.

The purpose of this document is to define a reproducible integration of an autonomous haulage system (AHS) and a fleet management system (FMS) to avoid customized implementations at every site. It allows a supplier with a narrow product coverage but highly valuable core mining competencies to participate and deliver open-autonomy components in the overall autonomy technology stack.

This document defines a software API, an open-autonomy interface, that allows independent vendors to supply an FMS to dispatch an AHS autonomous fleet. The API allows for future innovations and a wide variety of implementations without requiring a modification to the protocol.

This document interface aims to create an API that will deliver:

— system wide source of truth digital map and machine positions on this map

— dispatch functionality of autonomous trucks to support material movement, fuelling, and parking.

— generate autonomous equipment production monitoring.

The scope for viable interoperability between the AHS and FMS is the API and that is the focus of this document.

Autonomous system and fleet management system interoperability

1 Scope

This document defines the interfaces required between the fleet management (FMS) and autonomous haulage (AHS) systems for dispatch of haul trucks and coordination of production information, including communication protocols, message structures, telemetry signals, map sharing and task assignments.

This document applies to surface mining. It specifies requirements and recommendations to achieve the following:

- realtime computer system communication;
- message definition and semantic;
- mine map sharing;
- truck dispatching;
- truck production monitoring.

This document does not address computer system authentication, authorization and cyber security. These methods and technologies are already covered by best practice IT deployments.

The specific requirements for safe operation of machines, including execution of task assignments issued by the FMS to the AHS rely on additional information that is agreed between the FMS and AHS supplier which is outside the scope of this document.

2 Normative references

The following documents are referred in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, on the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 8601 (all parts), *Date and time — Representations for information interchange*

ISO 11992-2, *Road vehicles — Interchange of digital information on electrical connections between towing and towed vehicles — Part 2: Application layer for brakes and running gear*

3 Terms, definitions and abbreviations

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

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at https://www.iso.org/obp
- IEC Electropedia: available at https://www.electropedia.org/

3.1 Terms and definitions

3.1.1
all-stop function
function that brings all autonomous machines under the operator’s supervision to a halted state when initiated

[SOURCE: ISO 17757:2019, 3.1.18, modified — The word "system" was replaced by "function".]

3.1.2
actor
software involved in communication over a connection, either providing a service or using a service

3.1.3
attribute
<JSON object> term attribute referencing to a key as opposed to an “object”

3.1.4
operator
person having control and responsibility for operating a machine or the autonomous haulage system

3.1.5
manual mode
mode of operation in which a machine is controlled by an *operator* (3.1.4) who is responsible for monitoring the surroundings and for safe operation of all machine controls

[SOURCE: ISO 17757:2019, 3.1.13]

3.1.6
pose
machine position and orientation (heading)

Note 1 to entry: The pose needs a standard frame of reference {0,0,0} for each machine so they are properly placed in space for any coordinate systems.

3.1.7
tray
<dumper body>that portion of a dumper which carries material

3.1.8
risk assessment
overall process comprising a risk analysis and a risk evaluation

[SOURCE: ISO 12100:2010, 3.17]

3.2 Abbreviations

AOZ	autonomous operating zone
API	application program interface
AHS	Autonomous Haulage System
FMS	fleet management system
GNSS	global navigation satellite system
GUID	globally unique identifier
IANA	Internet Assigned Numbers Authority

JSON	Java Script Object Notation
LV	light vehicle
UID	unique identification
TLS	transport layer security
SOP	Standard Operating Procedure
SSL	secure socket layer. Synonymous with TLS. TLS is the newer version of SSL
TUM	time usage model
wss://	web socket secure

4 Computer communication

The goals of the protocol design shall be

a) event based (no polling),

b) bi-directional,

c) cloud friendly (leverage web technologies),

d) rapid disconnection detection, and

e) events are sent immediately,

Standard IT infrastructure is not the primary focus of this document. This document does not specify:

- security,
- authentication,
- redundancy,
- scaling,
- encryption, and
- certificates.

These functions should be implemented to support the demand of each specific deployment. See Annex B for more details.

4.1 Session and transport layer

The client system shall use a TCP/IP websocket secure (wss://) to connect to services defined in this document.

Once the connection is established, then the state of each data models shall be shared (not necessarily synchronized) between the client and the server.

Once the data models are shared, then each system shall update the other system through the wss:// connection as events and states change.

4.1.1 WebSocket client and service behaviour

The following should be behaviours of the WebSocket client and server:

— WebSocket clients should implement an automatic re-connection to the WebSocket server if the connection is lost.

— WebSocket servers (listeners) should implement a configurable Ping in accordance to RFC6455, Section 5.5.2.

— WebSocket clients may implement a configurable Ping in accordance to RFC6455, Section 5.5.2.

NOTE Some WebSocket implementations will close the connection if there is no frame transferred for a certain amount of time. RFC 6455, Section 5.5.2 provides a Ping-Pong function to prevent premature closure.

4.2 Presentation layer

The presentation layer is defined by OSI, presentation layer - ISO/IEC 7498-1 (basic model).

4.2.1 Container, serialization, and encoding

The following specifications shall apply to container, serialization, and encoding (see Table 1):

Table 1 — Container, serialization and encoding

Layer	Specifications
Container	JSON ECMA-404
Serialization	English text
Encoding	UTF-8

4.2.2 JSON container

The JSON notation shall be used in English human readable form to represent serialized object.

4.2.3 NULL attribute

If an attribute is required in a message (see 7.5) but is un-instrumented on the machine (i.e., it does not exist physically) from the sender, then the attribute shall be set to ‘null’.

NOTE Null has a different semantic than 0 and different from an empty array. NULL means it is absent. An empty array means that at the moment the list is empty, but the array may grow beyond empty in the future. Semantically, a zero value does not imply unknown, broken or invalid unless otherwise stated in the specification.

4.2.4 Units

The exchanged attributes shall use a common set of units for the different measurements. The following units shall be used for value transmission (see Table 2):

Table 2 — Measurement units

Measurement	Unit
UTC Date time stamp	ISO 8601 (all parts)
Time	seconds
Distance	meter
Speed	km/h
Angle	degree
Temperature	Celsius

Table 2 *(continued)*

Measurement	Unit
Pressure	kPa
Mass	kg
Heading	degree

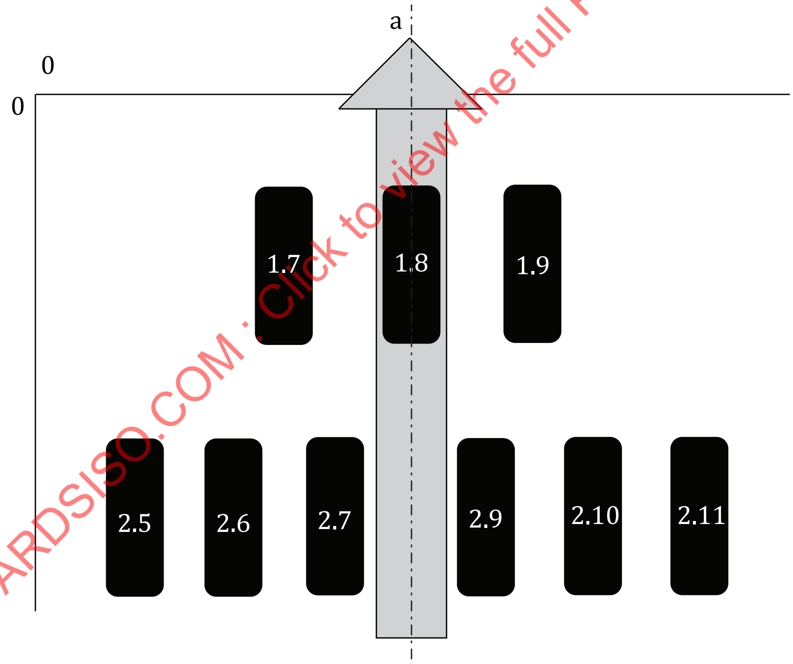
4.2.5 On machine component arrays

ISO 11992-2 shall be used to identify a particular tire/wheel and axle position. For consistency, the same numbering method shall be used to identify a particular machine component when there are multiple component instances on a machine.

Machine components shall be numbered according to their position on the machine. Represented as a 2D coordinate in the form {ROW, COLUMN}. Rows are starting from the front to the back of the machine in accordance to their position.

Row numbering shall start at 1.*x*, and increasing in increments of 1 (i.e. 1.*x*, 2.*x*, etc.). Objects to the left of the middle should count down from .7 and objects to the right of the middle should count up from .9. An odd cardinality of components (a middle component) shall be assigned ROW#**.8** (ROW# corresponding to the specific row number).

The following examples clarify when and how to use ROW#.8 for odd and even number of items; Figure 1 and Figure 2 a) and b) provide examples of component numberings.



[a] Forward driving direction.

Figure 1 — Example of tire number on 9-wheel rig (adapted from ISO 11992-2:2014)

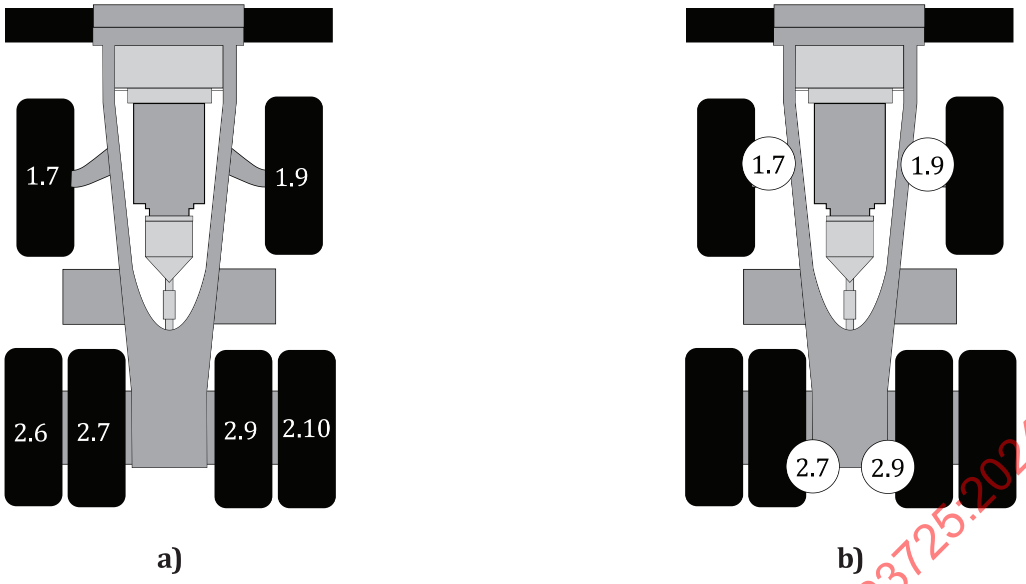


Figure 2 — Example of tire and strut numbering on a rigid dump truck

Figure 2 a) is numbering the tires of a rigid dump truck. The procedure is

- to orient the machine's front pointing upwards.
- All tires of the first row start with a "1.", the one to the left is "1.7" and the one on the right is "1.9". There is no "1.8" because there is no tire in the middle.
- All tires of the second row start with a "2.", the one to the most left is "2.6", then its right neighbour is "2.7". The 2 tires to the right of the middle are "2.9" and "2.10" is the right most.

Figure 2 b) is simpler as we're numbering the truck struts, use the same approach, the machine is oriented forward then follow the algorithm.

5 Messaging

5.1 Overview

The interface shall use asynchronous messaging and be compliant to the rule set in Clause 4, which defines the transport, container, encoding and behaviour.

Annex A contains a set of sequence diagram as examples of how the messages are expected to be used between the systems. Annex A should be consulted to understand how they are intended to be used under typical circumstances.

5.2 Event based messaging

- A message shall be sent when a state or measurement is changed within the monitored system. Continuously changing measurements shall be limited to a maximum message sending frequency.
- The maximum frequency shall be configurable on a per message type basis on the sending application during commissioning.
- Unchanged state messages shall not be sent periodically based on a timer.

5.3 Message header

5.3.1 General

All messages in conformity with to this document shall contain the message header as described in Table 3.

Table 3 — Message header

Attribute	Req. Level	Type	Exactly	Description
Protocol	shall	UKey	ISO23725	A unique human readable string used to identify that this message object is compliant with the ISO 23725 interface.
Version	shall	integer	1	The version of the protocol in use. This version shall change if there is a break in backward compatibility in: — New mandatory field in the header — Presentation layer
Timestamp	shall	ISO 8601 (all parts)	<YYYY-MM-DD>T <HH:mm:ss[.sss]>Z	UTC Time (+0 Zulu). When the data or event contained in the message was measured. Milliseconds are optional and should be added if meaningful. The timestamp is used to mark when the data or event occurred, not when the message was serialized or when it left the computer.
EquipmentId	should	GUID		If the message is pertaining to a specific machine, then the message shall include the Equipment Identification in the header. The GUID is a unique Identification number that represents one-and-only-one piece of equipment defined in the fleet.

NOTE Stuffing arrays of objects intended for different pieces of equipment in a single message is prevented by design. This is achieved by placing the equipment ID in the message header. This implies that communication targeted to multiple equipment, (but not the entire fleet) needs to be sent in separate messages.

5.3.2 Header example

A message that is meant for the AHS, and not machine specific, should have a header that follows this example of a non-machine specific message header:

Header of message that is not specific to a machine.

```
{
   "Protocol":"ISO23725",
   "Version": 1,
   "Timestamp": "2018-10-31T09:30:10.711Z"
}
```

A message that applies to a specific machine, in this example machine “2248d535-3daf-4a86-b1e1-4951a22beec6” should have a header that follows this example of a machine specific message header

Header of message that is specific to machineidentified by an equipment GUID.

```
{
   "Protocol": "ISO23725",
   "Version": 1,
   "Timestamp": "2018-10-31T09:30:10.511Z",
   "EquipmentId": "2248d535-3daf-4a86-b1e1-4951a22beec6"
}
```

5.4 Messages during connection outage

The actors participating in messaging should persist important messages during communication outages for a configurable time duration. When the communication is re-established, any persisted messages shall be sent over the re-established communication channel. The original timestamp of when the events occurred shall remain unchanged.

NOTE The main goal is to ensure that equipment data is not lost in the case there is an outage. It is up to the integrators to decide which message are important to be persisted to reach the desired application goal.

5.5 Out of standard message

5.5.1 General

Any message without a JSON attribute bearing a "Protocol":"ISO23725" attribute shall be IGNORED by the message interpreter. The message interpreter shall continue to operate normally after ignoring a message.

The websocket communication channel may be used to interweave out of standard messages that are not defined by this document, but supported by vendor specific implementations. All compliant message processors shall support the ability to discard unknown JSON messages that do not have a Protocol attribute or where the Protocol attribute is not equal to ISO23725 and shall continue to operate normally thereafter. The rejection of the message may be logged by the interpreter.

Out of standard messages that are sent on the same stream shall follow these conditions:

— The message shall be compliant with the formatting as defined in the presentation layer (see 4.2).

— There shall NOT be any attribute named “Protocol” that equals to "ISO23725".

5.5.2 Out of standard examples

Two valid examples of what could consist “out of standard” messages are provided as follows:

```
{
  "MySpecificMessage": "The Blue Turtle",
  "MySpecificData": "Too Secret to be public"
}
```

A site specific extension to a standard message

```
{
  "Protocol":"MyCustomizedStandard",
  "Timestamp": "2018-10-31T09:30:10.43.711Z",
  "GUID": "2248d535-3daf-4a86-b1e1-4951a22beec6",
  "BetterMachinePosition": {
    "Heading": 211,
    "Latitude": 49.176854678267288,
    "Longitude": -123.0718674232,
    "Elevation": 175.23,
    "LatitudeAccuracy": 0.3164,
    "LongitudeAccuracy": 0.2743,
    "HeightAccuracy": 0.58,
    "HeadingAccuracy": 2.1,
    "Speed": 0,
    "PDOP": 2.75,
    "ReceiverModel": "Ublox",
    "GyroUsed": "yes"
  }
}
```

5.6 Common attribute enumeration

5.6.1 General

This clause will define a common set of enumerated values and their semantic that can be assigned to message attributes.

5.6.2 Actor

The following actor enumeration shall be used when referencing a specific actor in a message (see Table 4).

Table 4 — Actor enumeration

Actor Enumeration	Meaning
AHS	The whole self driving functionality composed of onboard and possibly back office services.
FMS	The whole Fleet Management functionality
MAP	The service sharing the information describing the road map. Including road segments, intersections and open area.

5.6.3 Action request

Table 5 — Action request enumeration

Action Enumeration	Meaning
Apply	Request a service provider for an action to be executed
Remove	Request a service provider for an action to be halted

5.6.4 Action response

Table 6 — Action response enumeration

Enumeration	Meaning
Accepted	Answer by the service provider that the requested action will be executed
Rejected	Answer by the service provider that the requested action will NOT be executed

5.6.5 Action status

Table 7 — Action status enumeration

Enumeration	Meaning
On	The action is being executed
Off	No action executing

5.6.6 Autonomy mode

Table 8 — Autonomy mode enumeration

Enumeration	Meaning
Manual	The machine is under human control
Autonomous	The machine is under computer control
TransitionToAutonomous	A human is transitioning the machine control to the computer in a safe manner
TransitionToManual	The computer is transitioning the machine control to a human in a safe manner
AllStop	The machine is under computer control, but immobilized. Remote human intervention will be required to allow machine movement.
EmergencyStop	The machine is now under human control and all fail-safe, emergency stopping, physical controls are engaged on the machine.
RemoteControl	The machine is under human control, where the human is usually not in line of sight of the machine. The human controls the machine through computers and a communication channel.

Each autonomy provider has their unique way to safely transition a machine from manual mode to autonomous mode. During integration, the autonomy supplier shall provide a document to the end-user mapping their implementation specific modes to the ones that best fit the meaning description above.

EXAMPLE If there are multiple states within the TransitionToAutonomous, the autonomy supplier would document that to say "TransitionToAutonomous" = "Machine Selection" + " Machine Authentication" + "Button Pressing" + "Move to safe location" or any dominant state. Another example, if AllStop is dominant on a RemoteControl state, the autonomy provider needs to document that while in AllStop, a person in Remote control would lose control of the machine and the computer would keep it immobilized.

6 System messaging

6.1 General

System messages shall not have an `EquipmentId` attribute in their header.

6.2 Fleet definition message

6.2.1 General

The fleet definition message shall be the first message to be sent by both actors upon connection.

Actors shall be able to receive and process a new fleet definition message with updated content at runtime. This message shall be re-sent by an actor when the list of machines monitored by the actor is changed. The list can change by adding, deleting or modifying a machine. Both actors shall use the same GUID and use an equivalent description for the same physical machine so the semantic is preserved across both systems.

On fleet change, only the actor owning the list that was modified shall send an updated fleet definition message.

Actors shall send and receive the fleet definition message in accordance to Table 9.

Table 9 — Fleet definition message initiator and trigger

	AHS to FMS	FMS to AHS
Trigger: at connection	Yes	Yes
Trigger: runtime change	Yes	Yes

NOTE The purpose of this message is to catch important AHS–FMS fleet discrepancies and have end-users complete the configuration of the machine definition in the respective systems with their respective tools. This message is primarily to ensure that both systems are aware of the active machines that the other system manages.

The purpose is not to have a single synchronized source of Equipment definition. Rather a means to automatically monitor and alert a person when there are important discrepancies between the two systems. Each systems has a very distinct data model defining the machine attribute it requires per machine definition, there is no common data model that will covers all actor's needs.

Each system has their own proprietary machine data model that is configured with their respective native tools. The FleetDefinition is a simplified information exchanged to ensure that people managing both systems will be able to rapidly identify that they are talking about the same physical machine in both systems. The fleet definition can be validated by people because new autonomous equipment is rarely commissioned.

The content of a fleet definition message shall comply with (see Table 10).

Table 10 — Fleet definition structure

Member	Req. Level	Type	Description	Example
FleetDefinitionV2	shall	Object	Identifies this structure as a Fleet Definition	
AHSId	shall	GUID	A unique “key” that represents an AHS instance. This key is static across session and used for the life of the AHS instance it represents.	5318e44c-e9f0-42e2-9965-a4a1ff364f4b
Equipment	shall	Array of objects	Identifies this as a collection of equipment objects. The array can be empty but shall not be null.	"Equipment": [ ]
EquipmentId	shall	GUID	The unique static “key” used in other messages to reference this piece of equipment.	0c83193f-8772-446a-89c0-a3977e282b8a
HID	shall	String	Human readable label as defined in the source system. Only meant to be meaningful to people.	LV033, “Bucyrus 05”
Type	shall	Enumeration	Describes the equipment category	HaulTruck, Shovel
OEM	shall	String	Original Equipment Manufacturer. The brand name that is stamped all over the equipment.	Honda
Model	shall	String	OEM model number / name	Civic, 793D

Table 10 *(continued)*

Member	Req. Level	Type	Description	Example
Autonomous	shall	Boolean	Autonomy capable	true, false
Length	shall	Decimal	The distance in meters from the front most to the rear most of the equipment	4,56
Width	shall	Decimal	The distance in meters from the left most to the right most of the equipment	1,43

NOTE Any future editing of the fleet, deleting or marking a machine as inactive, will simply have the machine disappear from the new equipment list in the fleet definition message. There is no delete command.

6.2.2 Fleet ID

The fleet id shall be unique amongst all fleets connecting to one FMS instance.

The fleet id shall be immutable across connection sessions and system reboot.

A fleet definition shall have exactly one Fleet id associated with it.

NOTE Be mindful that the set of equipment in this fleet will vary over time as machines are put in and out of commission. The fleet ID allows the actors to implement these use cases:

- Allow multiple different AHS fleet to connect to a single FMS. The FMS will use the Fleet Id to differentiate between the different AHS actors.
- Single AHS instances to share multiple fleets with a single FMS. For instance, when fleets are operated independently in different pits or sites.
- Allow for a single AHS to connect multiple concurrent sessions to the FMS and provides the FMS with a way to identify that these multiple sessions are aggregated and "act as one" and that they control the same fleet.

6.2.3 Equipment type

Only a valid enumeration contained in Table 11 shall be used as an equipment type.

Table 11 — Machine type enumeration

Machine Type Enumeration

```
{
 HaulTruck,
 Shovel,
 Excavator,
 Loader,
 LightVehicle,
 WaterCart,
 Grader,
 FuelTruck,
 LubeTruck,
 Dozer,
 RubberTireDozer,
 Drill,
 Crusher,
 Scraper,
 BellyDumper,
 EmergencyVehicle,
 Ambulance,
 Dragline,
 SurfaceMiner,
 Bus,
 Train,
 Trailer
}
```

NOTE Equipment to equipment interactions are defined by the machine type and their capabilities, hence the need to identify and define the types of machines in the system. An autonomous truck will cross the path of other machines in the pit and could behave differently (slowdown, keep away or stop) depending on the machine type, model and if it's manned or autonomous.

6.2.4 Autonomous

This attribute shall be set to true if the machine can be used autonomously by an AHS otherwise it shall be set to false.

6.2.5 Equipment length and width

The length shall be the distance in meters from the front most to the rear most of the equipment.

The width shall be the distance in meters from the left most to the right most of the equipment.

NOTE The purpose of the equipment length and width are there so the systems have a general footprint of the equipment on the road. Each system is independently responsible for the level of data accuracy and integrity required for their application.

6.2.6 Fleet definition example

This is an example of what a very small fleet message could look like:

A Fleet with one truck and one light duty vehicle.

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10.43.511Z",

 "FleetDefinitionV2": {
  "AHSId": "5318e44c-e9f0-42e2-9965-a4a1ff364f4b",
  "Equipment": [
   {
    "EquipmentId": "0c83193f-8772-446a-89c0-a3977e282b8a",
    "HID": "HT042",
    "Type": "HaulTruck",
    "OEM": "ETF",
    "Model": "Virtual-F",
    "Autonomous": true,
    "Length": 14.57 ,
    "Width": 9.02
   },
   {
    "EquipmentId": "2248d535-3daf-4a86-b1e1-4951a22beec6",
    "HID": "LV033",
    "Type": "LightVehicle",
    "OEM": "Ford",
    "Model": "F350",
    "Autonomous": false,
    "Length": 4.67 ,
    "Width": 2.52
   }
  ]
 }
}
```

An empty fleet during commissioning.

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10.43.511Z",

 "FleetDefinitionV2": {
  "AHSId": "5318e44c-e9f0-42e2-9965-a4a1ff364f4b",
  "Equipment": [ ]
 }
}
```

6.3 Machine all-stop

6.3.1 Optional feature

The machine All-Stop functionality should only be enabled if a risk assessment has shown that the all-stop integration is safe and desired in the final solution.

If the machine All-Stop functionality is enabled, then these messages shall be implemented:

a) MachineAllStopV1;

b) MachineAllStopRequestV1;

c) MachineAllStopResponseV1.

If machine All-Stop is disabled, then these messages above shall not be sent.

While the All-Stop is active the whole autonomous fleet

— shall remain under computer control,

— shall be immobilized,

To allow first responders to drive safely in the AOZ and reach the emergency as fast as possible without negotiating the road with moving autonomous machines.

NOTE The machine all-stop function is not an emergency stop. It is a best effort function only and it is not fail-safe. The all-stop is not to be used as a means to stop an autonomous machine that went rogue or that is about to collide with an object. A different system will need to be used for these use cases. Therefore a risk assessment is required to properly implement the Machine All-Stop.

6.3.2 Scope of effect

The machine all-stop functionality shall affect all fleets connected to the FMS. The All-Stop is applied to each autonomous machine in each fleet.

6.3.3 Machine all-stop status message

The machine All-Stop status shall be sent from the AHS to the FMS at connection time and immediately when it changes as per Table 12.

Table 12 — Messages initiator and trigger

	AHS to FMS	FMS to AHS
Trigger: at connection	Yes	No
Trigger: runtime change	Yes	No

The All-Stop status {On,Off} shall not be used to determine if autonomous machines are immobilized.

The Machine all-stop status message shall follow the structure described in Table 13.

Table 13 — Machine all-stop status structure

Member	Req. Level	Type	Description	Example
MachineAllStopV1	shall	Object	Identifies this object as an: All-Stop Status	
ModeAllStop	shall	Action Status enumeration	States the status of the All-Stop mode for the fleet controlled by AHS. On means that it is active and Off is deactivated.	“On” or “Off”
ModeAllStopId	shall	GUID	Value of the GUID comes from: — The last GUID inside the MachineAllStopRequest message of the FMS that changed the status. — A random GUID generated by AHS if the activation was not triggered by the FMS. (when triggered by the native AHS application) — A random GUID generated by AHS when a new connection is established and All-Stop is active. — Null if All-Stop is deactivated on initial connection.	0c83193f-8772-446a-89c0-a3977e282b8a

6.3.3.1 ModeAllStop

The value of the mode all-stop shall be Off in normal AHS Operation. The value of the mode all-stop shall be On if the All-Stop mode is active on the AHS.

6.3.3.2 ModeAllStopId

The value of the mode all-stop id shall be the latest GUID value used during the latest ModeAllStop status change.

The source of the GUID shall be the latest of:

- FMS side generated GUID provided inside the MachineAllStopRequest message from the FMS
- AHS side generated GUID provided by AHS when the activation was triggered on AHS.
- AHS side generated GUID provided by AHS when a new connection is established and All-Stop is already active.

6.3.4 Machine all-stop request message

The message request to Apply or Remove an All-Stop shall be sent from the FMS to the AHS when initiated by an FMS user. Table 14 specifies which actor shall send the message and when this message is expected.

Table 14 — Messages initiator and trigger

	AHS to FMS	FMS to AHS
Trigger: at connection	No	No
Trigger: runtime change	No	Yes

NOTE The dispatcher is normally the centre of all voice communication for the fleet, this functionality allows for a dispatcher to initiate the safe immobilization of the autonomous fleet when the dispatcher is made aware that first responders will need to enter the AOZ. The standard operation procedure (people, process, guards and tools) of how the All-Stop function is operated at site is site specific and out of scope.

The Machine all-stop request message shall follow the structure described in Table 15.

Table 15 — Machine all-stop request structure

Member	Req. Level	Type	Description	Example
MachineAllStopRequestV1	shall	Object	Identifies this object as an: All-Stop Request	
ModeAllStop	shall	Action Request enumeration	Field that specifies the type of request from the FMS: { Apply, Remove }	“Apply”,”Remove”
ModeAllStopId	shall	GUID	A GUID specific to this request transaction. The Response to this request will be matched using this GUID.	“0c83193f-8772-446a-89c0-a3977e282b8a”

6.3.4.1 ModeAllStop

The FMS shall set this attribute to Apply to request to enable an All-Stop; alternatively set it to Remove to request that the All-Stop be lifted.

6.3.4.2 ModeAllStopId

The FMS shall generate a new GUID for each new request message. The FMS shall use the GUID to match the AHS reply to know the outcome of the request.

NOTE The GUID lifespan is only for a single request-response between the client and the server. Two different GUIDs need to be used when requesting to apply and when requesting to remove the All-Stop as it constitutes 2 different request-response. Using the same GUID for multiple requests may yield unexpected behaviours and is prohibited. The GUID is not used to link the "Apply" and "Remove" requests; but rather link the Request, the Reply and the source of change in the Status message.

6.3.5 Machine all-stop response message

The message response to the request shall be sent from the AHS to the FMS, moments after the FMS has sent an all-stop request see Table 17.

Table 16 specifies which actor shall send the message and when this message is expected.

Table 16 — Messages initiator and trigger

	AHS to FMS	FMS to AHS
Trigger: at connection	No	No
Trigger: runtime change	Yes	No

Table 17 — Machine all-stop response structure

Member	Req. Level	Type	Description
MachineAllStop ResponseV1	shall	Object	Identifies this object as an: All-Stop Request
Response	shall	Enum	Consult Table 6. The response indicates if AHS will comply with the FMS request.
ModeAllStopId	shall	GUID	The GUID corresponding to the specific FMS request instance.
Detail	may	String	Localized Human readable message explaining in more details the AHS Response; especially in the case of a rejection.

6.3.5.1 Response

The value of response shall be either one of “Accepted” or “Rejected”. AHS shall set it to “Accepted” if AHS will execute the requested action, otherwise it shall be set to “Rejected”.

6.3.5.2 ModeAllStopId

The value of ModeAllStopId shall be a copy of the GUID that was sent by the FMS in the Request message body that AHS is answering.

6.3.5.3 Detail

A localized human readable string may be provided to supplement the AHS response. The string shall not be longer than 1 024 bytes with an UTF-8 encoding.

NOTE To respect the specific AHS design, SOP and safety; the AHS implementation can always reject the request to remove the All Stop (by design).

6.3.6 Machine all-stop examples

Below are examples to help understand a few use cases of this message:

All Stop Status is Off. The normally expected state.	Sent by AHS

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10.43.511Z",

 " MachineAllStopV1": {
  "ModeAllStop": "Off",
  "ModeAllStopId": "34f388ec-09e9-4d95-9965-a4a1ff364f4b"
 }
}
```

A request to apply the All-Stop from the FMS.	Sent by FMS

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10.43.511Z",

 "MachineAllStopRequestV1": {
  "ModeAllStop": "Apply",
  "ModeAllStopId": "0e45793f-f1d0-420b-a1fa-85e3f42a7219"
 }
}
```

Positive answer from AHS for the All-Stop request from the FMS.	Sent by AHS

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10. 45.812Z ",

 "MachineAllStopResponseV1": {
  "Response": "Accepted",
  "ModeAllStopId": "0e45793f-f1d0-420b-a1fa-85e3f42a7219",
  "Detail": "Sure can do, 27 haulers are stopping."
 }
}
```

All Stop Status is ON. While there is an emergency in the AOZ.	Sent by AHS

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2020-06-06T09:30:10.43.725Z",

 " MachineAllStopV1": {
  "ModeAllStop": "On",
  "ModeAllStopId": "0e45793f-f1d0-420b-a1fa-85e3f42a7219"
 }
}
```

A request to remove the All-Stop from the FMS.	Sent by FMS

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10.43.511Z",

 "MachineAllStopRequestV1": {
  "ModeAllStop": "Remove",
  "ModeAllStopId": " 34f388ec-09e9-4d95-9866-4eec876b37b2"
 }
}
```

A negative answer from AHS for the All-Stop remove request.	Sent by AHS

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10.45.812Z",

 "MachineAllStopResponseV1": {
  "Response": "Rejected",
  "ModeAllStopId": "34f388ec-09e9-4d95-9866-4eec876b37b2",
  "Detail": "The FMS is not allowed to remove the All Stop on this AHS version."
 }
}
```

7 Generic machine telemetry

7.1 Message content

Machine telemetry messages shall have an `EquipmentId` attribute in their header.

7.2 Data streams

This section defines messages that are related to on-machine data streams for sensors and actuators. Because sensor readings are constantly changing, they can't be sent on change. These messages will typically be time constrained with a period by setting a maximum update frequency.

This document does not prescribe the maximum or minimum frequency at which these messages are to be sent. The system integration team will determine how these streams will be used and what is the appropriate period to send each message. The integration team shall take into consideration the wireless bandwidth utilization, server resources, database space requirement and the end goal of the data collection.

7.3 Machine position

All actors shall send the Machine Position changes in their fleet. All Positions shall use an absolute earth frame of reference (Latitude, Longitude and elevation).

7.3.1 Machine coordinate system and frame of reference

The machine coordinate system shall be 3D cartesian using meters. The machine coordinate system's frame of reference of {0,0,0} of machines shall be

- {0,0,0} At front most protrusion, ground level for wheeled and tracked machines (because this point can be physically reached so can be physically surveyed) with right hand axis Y pointing forward. See Figure 13.
- {0,0,0} At the coupler on the tongue of an attachment (trailer).
- {0,0,0} At centre of rotation for swinging (rotating) machines (easily surveyed with HP GNSS onboard). With right hand axis Y pointing towards ground engagement tool.
- {0,0,0} At centre pin of articulated machines with right hand axis Y pointing forward.

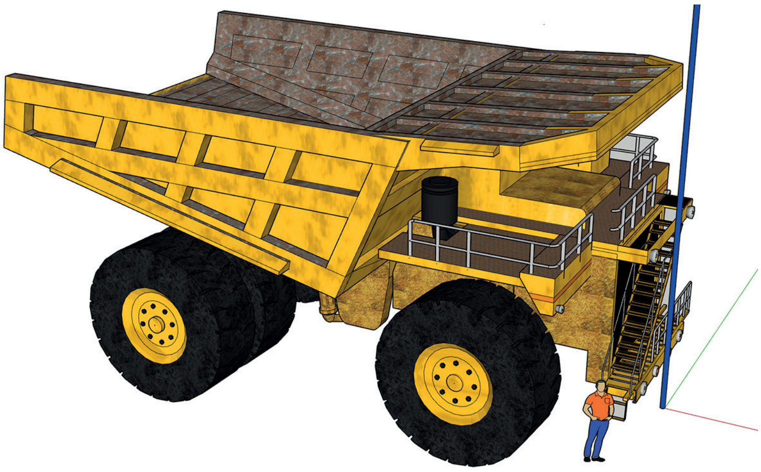


Figure 3 — Example of machine frame of reference

Trailers and articulated machines shall send multiple machine positions to completely describe their pose (see attachment structure). Each machine segment will be modelled as having its own machine identity and position. This is especially important when attachments are moved from machine to machine.

7.4 Machine position message

Table 18 specifies which actor shall send the message and when this message is expected.

Table 18 — Messages initiator and trigger

	AHS to FMS	FMS to AHS
Trigger: at connection	Yes	Yes
Trigger: runtime change	Yes	Yes

For each machine defined in an actor's Fleet Definition, the actor shall report the last known machine position at connection time. When the position of a machine changes, the actor shall send a new machine position message for that machine. The timestamp associated with the position shall be the time when the position was measured. The latest machine position message shall not be sent sooner than the minimum update period. Unchanged machine position may be sent. If the position of a machine is unknown by the actor, then the MachinePosition object shall be set to null.

NOTE Setting the position to 'null' is the proper way to signify that the position in UNKNOWN. Setting the position to 0,0,0 is not. If there is a last known position, however old, then it's better to use the last known position and set the timestamp to when the last known position in the MachinePosition::TimeStamp attribute.

There is no mandatory update frequency. Table 19 contains update periods used in the industry that may serve as a guideline.

Table 19 — Update periods for machine position

	Update period			Frequency
Maximum	1 000 ms	=	Minimum	1 Hz
Nominal	250 ms	=	Nominal	4 Hz
Minimum	100 ms	=	Maximum	10 Hz

When a machine position message is sent, it shall follow the structure defined in Table 20

Table 20 — Machine positions structure

Member	Req. Level	Type	Unit	Description	Example
MachinePositionV1	shall	Object	-	Identifies this object as a: Machine Position	
Heading	shall	integer	degrees	Degrees, WGS84 Compass heading in degrees with range of 0-359	67
Latitude	shall	decimal	degrees	WGS84 Latitude with a minimum of 6 decimals	49,176 854
Longitude	shall	decimal	degrees	WGS84 Longitude with a minimum of 6 decimals	-123,071 800
Elevation	shall	decimal	Meter	WGS84 Elevation in meters with 2 decimals	175,23
LatitudeAccuracy	shall	decimal	Meter	Latitude error @ 1 sigma probability	0,06
LongitudeAccuracy	shall	decimal	Meter	Longitude error @ 1 sigma probability	0,07
ElevationAccuracy	shall	decimal	Meter	Elevation error @ 1 sigma probability	0,12
HeadingAccuracy	shall	decimal	degrees	Heading error @ 1 sigma probability	1,7
Speed	shall	integer	km/h	Ground speed of the machine	52
Timestamp	shall	UTC Time	-	The timestamp is the time the position was measured.	
Attachment [ ]	may	array	-	Attachment(s) connected to the main tractor if it applies.	

An attachment structure is a machine position with an equipment identification GUID as described by Table 21.

Table 21 — Attachment structure

AttachmentV1 member	Req. Level	Type	Unit	Description
Equipment ID	shall	GUID	-	Unique Equipment ID in the fleet
Heading	shall	integer	degrees	Degrees, WGS84 Compass heading in degrees with range of 0-359
Latitude	shall	decimal	degrees	WGS84 Latitude with a minimum of 6 decimals
Longitude	shall	decimal	degrees	WGS84 Longitude with a minimum of 6 decimals
Elevation	shall	decimal	meter	WGS84 Elevation in meters with 2 decimals
LatitudeAccuracy	shall	decimal	meter	Latitude error @ 1 sigma probability
LongitudeAccuracy	shall	decimal	meter	Longitude error @ 1 sigma probability
ElevationAccuracy	shall	decimal	meter	Elevation error @ 1 sigma probability
HeadingAccuracy	shall	decimal	degrees	Heading error @ 1 sigma probability
Speed	shall	integer	km/h	Ground speed of the machine

7.4.1 Machine position examples

A parked machine.

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10.43511Z",
 "EquipmentId":"2248d535-3daf-4a86-b1e1-4951a22beec6",

 "MachinePositionV1":
 {
  "Heading": 211,
  "Latitude": 49.176854,
  "Longitude": -123.0718,
  "Elevation": 175.23,
  "LatitudeAccuracy": 0.31,
  "LongitudeAccuracy": 0.27,
  "HeightAccuracy": 0.58,
  "HeadingAccuracy": 2.1,
  "Speed": 0
 }
}
```

Segmented machine or single attached trailer providing multiple machine positions.

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10.43511Z",
 "EquipmentId":"2248d535-3daf-4a86-b1e1-4951a22beec6",

 "MachinePositionV1":
  {
  "Heading": 211,
  "Latitude": 49.176854,
  "Longitude": -123.0718,
  "Elevation": 175.23,
  "LatitudeAccuracy": 0.31,
  "LongitudeAccuracy": 0.27,
  "HeightAccuracy": 0.58,
  "HeadingAccuracy": 2.1,
  "Speed": 52

  "AttachmentV1":
  [
   {
   "EquipmentId": "e66c739e-b5f3-41f2-8bbb-5a29875ae70d",
   "Heading": 220,
   "Latitude": 49.176812,
   "Longitude": -123.071783,
   "Elevation": 175.18,
   "LatitudeAccuracy": 0.31,
   "LongitudeAccuracy": 0.27,
   "HeightAccuracy": 0.58,
   "HeadingAccuracy": 2.1,
   "Speed": 52
   }
  ]
  }
}
```

OLD (about 10 h) machine position when an event triggered this message transmission.

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10.43511Z",
 "EquipmentId":"2248d535-3daf-4a86-b1e1-4951a22beec6",

 "MachinePositionV1": {
  "Timestamp": "2018-10-30T23:11:53.963Z"",
  "Heading": 211,
  "Latitude": 49.176854,
  "Longitude": -123.0718, "Elevation": 175.23 ,
  "LatitudeAccuracy": 0.31,
  "LongitudeAccuracy": 0.27,
  "HeightAccuracy": 0.58,
  "HeadingAccuracy": 2.1,
  "Speed": 0
 }
}
```

Unknown machine position.

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10.43511Z",
 "EquipmentId":"2248d535-3daf-4a86-b1e1-4951a22beec6",

 "MachinePositionV1": null
}
```

7.5 Basic machine health message

The machine basic health information message should be sent by AHS. The basic health information message shall include the structure defined in Table 23.

Table 22 specifies which actor may send the message and when this message is expected.

Table 22 — Messages initiator and trigger

	AHS to FMS	FMS to AHS
Trigger: at connection	No	No
Trigger: runtime change	Yes	No

The purpose of this message is to allow fleet operators to keep the machines operational and avoid premature failures. Dispatchers, maintenance crew and FMS Dispatching will use:

— Fuel level to decide when is most optimal to refuel a machine. This depends on how far and busy is the fuel island or if a fuel truck is close to that area.

— Machine hours/engine hours: is used as a crude way to determine when a truck needs to come back for maintenance.

— Prime mover temperature to see if the hot weather is affecting the equipment.

- Tire pressure and temperature to make sure the tires are not used outside their specifications. The machine might be sent to maintenance if the pressures are too low, and it may be loaded with less material or told to slowdown if the tires are overheating.

Table 23 — Basic machine health structure

Member	Req. Level	Type	Unit	Description	Example
BasicMachineHealthV2	shall	Object	-	Identifies this structure as a Basic Machine Health	
FuelLevel	shall	integer	%	As defined by SAEJ1939 SPN-96 and 38	75
TotalMachineHour	should	decimal	h	As defined by SAEJ1939 SPN-246. Single decimal	7 451,1
TotalEngineHour	shall	decimal	h	As defined by SAEJ1939 SPN-247. Single decimal	3 756,8
Tire	should	Object Array	-	A list of an object containing tire properties	
Array Of [ ]					
TirePosition	shall	string		As defined by 4.2.5	
TirePressure	shall	integer	kPa	As defined by SAEJ1939 SPN-241	271
TireTemperature	shall	integer	°C	As defined by SAEJ1939 SPN-242	47

Tire pressure and Tire temperature are arrays because there are multiple instances per machine. Tire measurements will be identified by the standard component identification as defined in 4.2.5.

There is no mandatory update frequency. The appropriate frequency will be determined during the system integration and will be set to achieve the desired site goal.

NOTE As an example, a simple implementation could be for machines to report their health every ***5*** min. A more complex method could be to report when one of these conditions occur:

- a change of fuel level of more than 2,5 %;
- an increase of 1 h on the engine hour;
- a change of 10° in the prime mover temperature;
- a change of 70 kPa in a tire pressure;
- no update since 30 min.

7.5.1 Basic machine health message example

Machine that is fully instrumented.

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10.43.511Z",
 "EquipmentId": "2248d535-3daf-4a86-b1e1-4951a22beec6",

 "BasicMachineHealthV2": {
  "FuelLevel": 74,
  "TotalMachineHour": 4876.4,
  "TotalEngineHour": 735.1,
  "PrimeMoverTemp": 147,
  "Tire": [
   {“TirePosition”:"1.7", “TirePressure”: 303, “TireTemperature”: 39},
   {“TirePosition”:"1.9", “TirePressure”: 297, “TireTemperature”: 39},
   {“TirePosition”:"2.6", “TirePressure”: 335, “TireTemperature”: 42},
   {“TirePosition”:"2.7", “TirePressure”: 334, “TireTemperature”: 41},
   {“TirePosition”:"2.9", “TirePressure”: 329, “TireTemperature”: 42},
   {“TirePosition”:"2.10", “TirePressure”: 332, “TireTemperature”: 43}
   ]
 }
}
```

Machine with minimal instrumentation.

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10.43.511Z",
 "EquipmentId": "2248d535-3daf-4a86-b1e1-4951a22beec6",

 "BasicMachineHealthV2": {
  "FuelLevel": 25,
  "TotalMachineHour": 4876.4,
  "PrimeMoverTemp": null
 }
}
```

8 Generic machine messaging

8.1 Message content

Generic machine messages shall contain an `EquipmentId` attribute in their header.

Generic machine messages shall contain a machine position structure in their body, as defined in Table 20.

8.2 Machine autonomy mode message

AHS shall send machine autonomy mode messages.

Table 24 specifies which actor is allowed to send the message and when this message is expected.

Table 24 — Messages initiator and trigger

	AHS to FMS	FMS to AHS
Trigger: at connection	Yes	No
Trigger: runtime change	Yes	No

Table 25 — Machine autonomy mode structure

Member	Req. Level	Type	Description	Example
MachineAutonomyModeV1	shall	Object	Identifies this structure	MachineAutonomyModeV1
AutonomyMode	shall	enumeration		Manual

Table 26 — Machine autonomy mode message example

Machine Autonomy mode

```
{
   "Protocol":"ISO23725",
   "Version": 1,
   "Timestamp": "2018-10-31T09:30:10.351Z",
   "EquipmentId": "2248d535-3daf-4a86-b1e1-4951a22beec6",

   "MachineAutonomyModeV1": {
      "AutonomyMode": "Autonomous"
   },
   "MachinePositionV1": {
     "Heading": 211,
     "Latitude": 49.176854,
     "Longitude": -123.0718,
     "Elevation": 175.23,
     "LatitudeAccuracy": 0.31,
     "LongitudeAccuracy": 0.27,
     "HeightAccuracy": 0.58,
     "HeadingAccuracy": 2.1,
     "Speed": 0
   }
}
```

8.3 Machine dispatch availability

8.3.1 General

AHS shall send machine dispatch availability messages.

Table 27 specifies which actor is allowed to send the message and when this message is expected.

Table 27 — Message initiator and trigger

	AHS to FMS	FMS to AHS
Trigger: at connection	Yes	No
Trigger: runtime change	Yes	No

8.3.2 Purpose

This message is used by AHS to inform the FMS which actor has dispatching authority over a machine in its fleet. AHS has sole authority to decide which actor can dispatch which machine in its fleet. The autonomous machine will only accept assignments from the actor that possesses dispatching authority. The AHS will set the DispatchingCustodian to AHS to stop the FMS from sending dispatching to the machine.

8.3.3 Machine Dispatch Availability message

The dispatch availability message shall include the structure defined in Table 28

Table 28 — Machine dispatch availability structure

Member	Req. Level	Type	Description	Example
MachineDispatchAvailabilityV1	shall	Object	Identifies this structure	
DispatchingCustodian	shall	enumeration	States which actor is the current Custodian. See Table 29	FMS
DispatchingNextCustodian	may	enumeration	States which actor will become the next Custodian. See Table 29	AHS
NextCustodianHandover	may	int	States approximatively when the next custodian change will occur in seconds.	300 0 would mean now, 300 would mean in 5 min

Table 29 defines how the system will interpret the meaning of the content of the message

Table 29 — Machine dispatch availability semantic

Current Custodian	Next Custodian	Time Estimate	Description
FMS	N/A	N/A	The FMS has dispatching authority over this machine. The FMS shall send dispatching instructions as soon as possible. This use case shall be implemented.
AHS	N/A	N/A	The AHS has dispatching authority over this machine. The machine will not accept instructions from an external dispatching source. This use case shall be implemented.
AHS	FMS	XXX	The FMS will receive dispatching authority over this machine in approximately XXX seconds. This use case may be implemented.
FMS	AHS	XXX	The FMS will loose dispatching authority over this machine in approximately XXX seconds. This use case may be implemented.

NOTE When a machine is in autonomous mode, that does not imply that it is available to be dispatched by the FMS. Machines in autonomous mode could be in testing or working under the dispatching authority of the AHS. This message is used to explicitly transfer the dispatching responsibility from the AHS to the FMS, or for AHS to take it back.

8.3.4 Trigger the FMS to re-send an assignment

When the FMS receives a MachineDispatchAvailability message

— with the DispatchingCustodian attribute equal to FMS, and

— without a DispatchingNextCustodian attribute,

then the FMS shall send an assignment message to the machine as soon as possible even if one was sent previously.

8.3.5 Impending custodian change

Optionally the AHS can warn the FMS of impending custodian change. The AHS may send a dispatch availability message with the DispatchingNextCustodian and NextCustodianHandover attributes set.

NOTE Notifying of an impending custodian change is a courtesy to the FMS that allows the FMS to optimize its short term planning. The FMS looks forward into the future to try to optimize the fleet's utilization. A truck being removed or added to the production fleet has an impact on the forecasted assignments. When the AHS can notify the FMS about machines becoming available or unavailable, this helps to optimize the overall production of the fleet.

The predicted handover time is an estimate so it is not expected to be exact. The time estimate is only an indicator that allows the FMS to better plan the next assignments. The machine does NOT automatically become FMS controlled once the delay is expired. An explicit MachineDispatchAvailability message will be sent by AHS when the custody change occurs.

8.3.6 Machine dispatch availability example

The AHS is informing the FMS that taking effect immediately, the FMS has DISPATCHING control of the machine ID 2248d535-3daf-4a86-b1e1-4951a22beec6, see Table 30.

Table 30 — FMS dispatching instruction to machine immediately

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10.43.511Z",
 "EquipmentId": "2248d535-3daf-4a86-b1e1-4951a22beec6",

 "MachineDispatchAvailabilityV1": {
  "DispatchingCustodian":"FMS"
  }

 "MachinePositionV1": {
  "Timestamp":"2018-10-31T09:30:10.43.511Z",
  "EquipmentGUID":"2248d535-3daf-4a86-b1e1-4951a22beec6",
  "Heading":"211",
  "Latitude":"49.176854",
  "Longitude":"-123.071800",
  "Elevation":"175.23",
  "LatitudeAccuracy":"0.31",
  "LongitudeAccuracy":"0.27",
  "HeightAccuracy":"0.58",
  "HeadingAccuracy":"2.1",
  "Speed":"0.0"
  }
}
```

The AHS is informing the FMS that taking effect immediately, see Table 31, the AHS is taking back the dispatching control from the FMS so the FMS can't expect the AHS machine to obey the current assignment, but the FMS can continue monitoring and reporting the truck's production.

Table 31 — AHS informing FMS that is it taking effect immediate

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10.43.511Z",
 "EquipmentId":"2248d535-3daf-4a86-b1e1-4951a22beec6",
 "MachineDispatchAvailabilityV1": {
   "DispatchingCustodian":"AHS"
 }
"MachinePositionV1": {
 "Timestamp":"2018-10-31T09:30:10.43.511Z",
 "Heading":"211",
 "Latitude":"49.176854",
 "Longitude":"-123.071800",
 "Elevation":"175.23",
 "LatitudeAccuracy":"0.31",
 "LongitudeAccuracy":"0.27",
 "HeightAccuracy":"0.58",
 "HeadingAccuracy":"2.1",
 "Speed":"0.0"
}
}
```

In this optional use case, the machine 2248d535-3daf-4a86-b1e1-4951a22beec6 is being mode-changed to autonomous and will be available in about 10 min, see Table 32.

Table 32 — Notify FMS that a new machine will become available in about 10 min

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10.43.511Z",
 "EquipmentId": "2248d535-3daf-4a86-b1e1-4951a22beec6",

"MachineDispatchAvailabilityV1": {
 "DispatchingCustodian":"AHS",
  "DispatchingNextCustodian":"FMS",
  "NextCustodianHandover":600
 },
 "MachinePositionV1": {
  "Heading": 211,
  "Latitude": 49.176854,
  "Longitude": -123.0718,
  "Elevation": 175.23,
  "LatitudeAccuracy": 0.31,
  "LongitudeAccuracy": 0.27,
  "HeightAccuracy": 0.58,
  "HeadingAccuracy": 2.1,
  "Speed": 0
 }
}
```

8.4 Machine diagnostic message

8.4.1 General

The machine diagnostic message shall be sent by AHS.

Table 33 specifies which actor is allowed to send the message and when this message is expected.

Table 33 — Messages initiator and trigger

	AHS to FMS	FMS to AHS
Trigger: at connection	Yes	No
Trigger: runtime change	Yes	No

8.4.2 Purpose

The purpose of the diagnostic message is for AHS to notify the FMS of any changes to the state of the autonomous machine, including:

- Cause of stoppage,
- Cause of performance degradation (slowdown),
- Caused externally by the environment (mining operations, weather, traffic),
- Caused internally by a malfunction or degradation of AHS, and
- Caused by a mechanical machine failure,

On receiving a diagnostic message, the FMS may then alert mining operations to tactically remedy the situation, and also review degradation trends in their time usage model to change and optimize their mining standard operating procedure.

The machine diagnostic message shall include the structure defined in Table 34

Table 34 — Machine diagnostic structure

Member	Req. Level	Type	Description
MachineDiagnosticV2	shall	Object	Identifies this structure
Array Of [ ]			
Component Id	shall	Integer	A component code number that identifies a subsystem monitored by the AHS.
Fault Code	shall	Integer	A number that matches a fault description for that component.

AHS shall concatenate all currently active faults present on the machine into a single message. AHS shall send a new message when the state of a fault changes: to active or to deactivated. AHS shall send a diagnostic message with an empty array of fault when the last fault is deactivated.

If a fault is not present in the diagnostic message, then it means that fault was not active at the time the message was sent. When the last fault is cleared, an empty fault list shall be sent to indicate that the machine is operating normally with no degradation.

8.4.3 Component identification number

The componentIds shall be defined by each AHS and should represent logical system components of the autonomous machine. Hardware and software are considered components. Example:{ ECM #1, Engine, VIMS, PLM, Obstacle Detection, Driving conditions, positioning sensor, lidar, driving rules, ...}.

8.4.4 Fault Code number

The fault codes shall be defined by each AHS and should represent a unique fault within the component identified by the parent component ID.

8.4.5 AHS documentation

The AHS version, component IDs and fault codes shall be documented and delivered by AHS with each release. The combination of { AHS Version + componentId + FaultCode } shall resolve to a uniquely identifiable documented fault.

8.4.6 Connection time

At connection time, if there are no active faults on a machine then the AHS shall send an empty fault array for each healthy machine. If active faults are present, then AHS shall send the concatenated list of active faults.

8.4.7 When the last fault clears

AHS shall send an empty Fault array when the last fault has cleared from the machine.

NOTE "MachineDiagnosticV2": `[ ]` is the proper way to encode an empty Fault array. Setting the MachineDiagnosticV2 attribute to NULL has a different semantic and is not supported.

The list below is non-exhaustive and represents examples of faults that should be considered as candidates for the diagnostic message.

- external causes:
  - obstacle detected;
  - proximity breached by a manned vehicle;
  - waiting for an assignment for the FMS;
  - dirty lens;
  - GNSS accuracy is inadequate;
- operational causes:
  - waiting at crusher;
  - waiting at shovel;
  - speed derated due to overloading;
  - tire Inspection;
- mechanical:
  - over temperature;
  - low tire pressure;
  - speed derated due to tire over temperature;
- network:
  - GNSS corrections lost;

  - Truck is out of radio coverage;
- weather:
  - speed derated due to Blizzard lidar low visibility;
  - speed derated due to Heavy rain slippery conditions.

8.4.8 Machine diagnostic message examples

The example in Table 35 represents a machine that has no active faults.

Table 35 — Machine diagnostic – no active faults

```
{
  "Protocol":"ISO23725",
  "Version": 1,
  "Timestamp": "2018-10-31T09:30:10.43.511Z",
  "EquipmentId": "2248d535-3daf-4a86-b1e1-4951a22beec6",

    "MachineDiagnosticV2": [],

    "MachinePositionV1": {
    "Heading": 211,
    "Latitude": 49.176854,
    "Longitude": -123.0718,
    "Elevation": 175.23,
    "LatitudeAccuracy": 0.31,
    "LongitudeAccuracy": 0.27,
    "HeightAccuracy": 0.58,
    "HeadingAccuracy": 2.1,
    "Speed": 0
  }
}
```

3 s later, the same equipment has 2 active faults, see Table 36.

Table 36 — Machine diagnostic – with 2 active faults example

```
{
  "Protocol":"ISO23725",
  "Version": 1,
  "Timestamp": "2018-10-31T09:30:10.46.911Z",
  "EquipmentId": "2248d535-3daf-4a86-b1e1-4951a22beec6",

    "MachineDiagnosticV2": [
      {"ComponentId": 101, "FaultCode": 386 },
      {"ComponentId": 3, "FaultCode":  8475}
    ],
    "MachinePositionV1": {
    "Heading": 211,
    "Latitude": 49.176854,
    "Longitude": -123.0718,
    "Elevation": 175.23,
    "LatitudeAccuracy": 0.31,
    "LongitudeAccuracy": 0.27,
    "HeightAccuracy": 0.58,
    "HeadingAccuracy": 2.1,
    "Speed": 0
  }
}
```

10 s later, all active faults are cleared, see Table 37.

Table 37 — Machine diagnostic – active faults cleared

```
{
  "Protocol":"ISO23725",
  "Version": 1,
  "Timestamp": "2018-10-31T09:30:10.57.021Z",
  "EquipmentId": "2248d535-3daf-4a86-b1e1-4951a22beec6",

    "MachineDiagnosticV2": [],

    "MachinePositionV1": {
    "Heading": 211,
    "Latitude": 49.176854,
    "Longitude": -123.0718,
    "Elevation": 175.23,
    "LatitudeAccuracy": 0.31,
    "LongitudeAccuracy": 0.27,
    "HeightAccuracy": 0.58,
    "HeadingAccuracy": 2.1,
    "Speed": 0
  }
}
```

9 Routing

Routing related messages shall not have an `EquipmentId` attribute in their header.

9.1 Map connectivity

AHS shall send Map Connectivity messages.

Table 38 specifies which actor is allowed to send the message and when this message is expected.

Table 38 — Messages initiator and trigger

	AHS to FMS	FMS to AHS
Trigger: at connection	Yes	No
Trigger: runtime change	Yes	No

9.1.1 Purpose

The purpose of the Map connectivity message is for the AHS to explicitly communicate to the FMS which road segment it can drive on and the order in which they can be traversed. The FMS will restrict the routes in its dispatching to only use the connections documented in this message. This message does not define any of the map geometry, but it references WayIds that are defined in an external shared map.

It's important to remember that the road connectivity is not interpreted by the geometry in the map, only through the explicit documentation of the connectivity message.

9.1.2 Map Connectivity message

The Map connectivity message shall include the structure defined in Table 39

Table 39 — Map connectivity structure

Member	Req. Level	Type	Unit	Description
MapConnectivityV1	shall	Object	—	Identifies this structure as a Map connectivity
Array Of [ ]				
WayId	shall	Integer	Unique ID	The ID of the current map object that is being described as connected to.
WayType	shall	Enum	{Road, Area}	The type of map object this is; Either an Area or a Road.
RoadProperty	Shall when Road	Object	—	See **Table 40**
AreaProperty	Shall when Area	Object	—	See **Table 41**
Connection	shall	array of WayId	Unique ID	The connection array is a list of other map objects that can be reached from the WayId attribute defined above. This connectivity is **unidirectional.** Defining a connection starting from the WayId attribute above to each individual map object in the connection list.

Table 40 — Road property structure

Member	Req. Level	Type	Unit	Description
Length	shall	Decimal	meter	The linear length of the road segment. Used by the FMS to calculate a seed theoretical travel time to drive this road segment.
Name	should	String		A human meaningful road name used by AHS.

Table 41 — Area property structure

Member	Req. Level	Type	Unit	Description
DefaultTask	shall	Enum	Machine Specific	The normal purpose or designation of the area. This indicated to the FMS what is the primary purpose of this area and what kind of tasks the dispatcher should be allowed to request in it.
Name	should	String		A human meaningful Areas name used by AHS.

The AHS shall build a list that includes all the map objects identified by a WayIds where it can operate autonomously. For each WayId in that list, the AHS shall list all the connections (WayIds) it can reach autonomously from that map object. The connections defined in the list are **unidirectional**. If a road segment can be traversed in both directions, then the other connection shall also be defined in the other WayId. The Map connectivity shall apply to the whole autonomous fleet defined by this connection.

NOTE In the case that an AHS controls multiple fleet size that can't share the same connectivity map, then the AHS should register with multiple connections each with its own fleet identification and its own Map Connectivity.

9.1.3 Clarification of unidirectionality of connections

The specification states that if road A is connected to road B and that the 2 road segments A and B can be driven in both directions, then the map connectivity message shall contain both connections:

- {A → can reach → B}, and
- {B → can reach → A}.

Neither statement implies the other. i.e. the statement "A connects to B" does not imply that "B connects to A". The statement "B connects to A" does not imply that "A connects to B".

9.1.4 Understanding the specification by example

Here's a simple abstracted map that we'll use to illustrate how the Map Connectivity message can be used by the AHS, depending on how the AHS wants to receive the dispatching instructions, see Figure 18. The AHS may use the level of details it wants. In this abstraction, we will use L, C and D to represent 3 areas instead of a real WayID integer. We will also use simple numbers to represent drivable roads or lanes instead of a real WayID integer. This is to simplify your understanding.

Road model

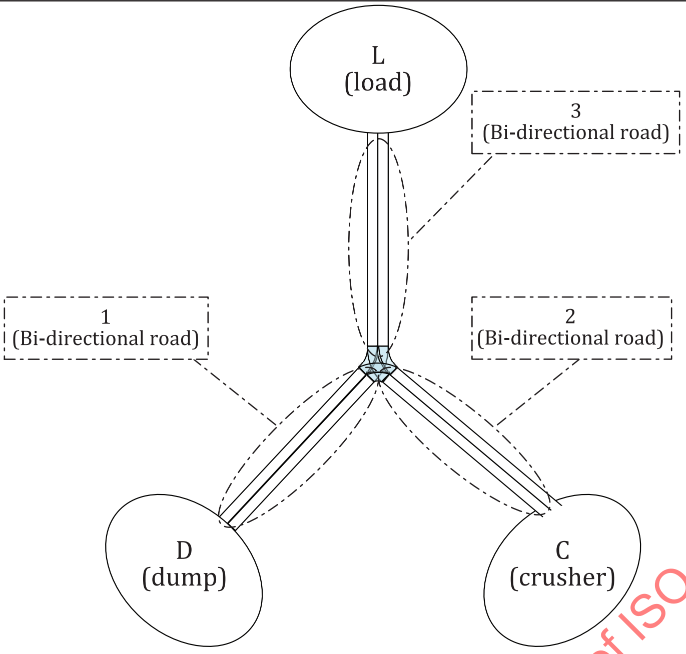


Figure 4 — Example of road level modelling

Dumping Area (D) is connected to

- (1)

Crusher Area (C) is connected to

- (2)

Loading Area (L) is connected to

- (3)

Segment (3) is connected to

- (L)
- (1)
- (2)

Segment (1) is connected to

- (D)
- (2)
- (3)

Segment (2) is connected to

- (C)
- (1)
- (3)

NOTE that C,D,L are representations of integer values and should not be used as-is.

```
"MapConnectivityV1": [
  {
  "WayId": D,
  "WayType": "Area",
  "AreaProperty":{
    "DefaultTask": "Offload",
    "Name": "Main Dump"
    }
  "Connection": [ {"WayId": 1} ]
  },
{
"WayId": C,
"WayType": "Area",
"AreaProperty":{
  "DefaultTask": "Offload",
  "Name": "Crusher"
  }
"Connection": [ {"WayId": 2} ]
},
{
"WayId": L,
"WayType": "Area",
"AreaProperty":{
  "DefaultTask": "Load",
  "Name": "Active Face"
  }
"Connection": [ {"WayId": 3} ]
},
{
"WayId": 1,
"WayType": "Road",
"RoadProperty":{
  "Length": 200.4,
  "Name": "Towards dump"
  },
"Connection": [
  {"WayId": D},
  {"WayId": 2},
  {"WayId": 1}
  ]
},
{
"WayId": 2,
"WayType": "Road",
"RoadProperty":{
  "Length": 194.8,
  "Name": "Crusher Avenue"
  },
"Connection": [
  {"WayId": C},
  {"WayId": 1},
  {"WayId": 3}
  ]
},
{
"WayId": 3,
"WayType": "Road",
"RoadProperty":{
  "Length": 202.3,
  "Name": "Towards Active face"
  },
"Connection": [
  {"WayId": L},
  {"WayId": 1},
  {"WayId": 2}
  ]
}
```

Dispatching Instructions from Loading to Dump

- (L)
- (3)
- (1)
- (D)

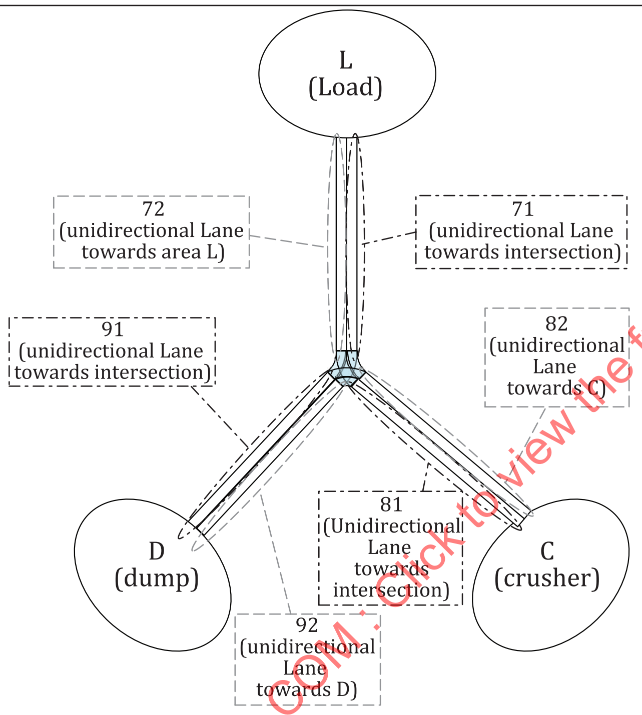
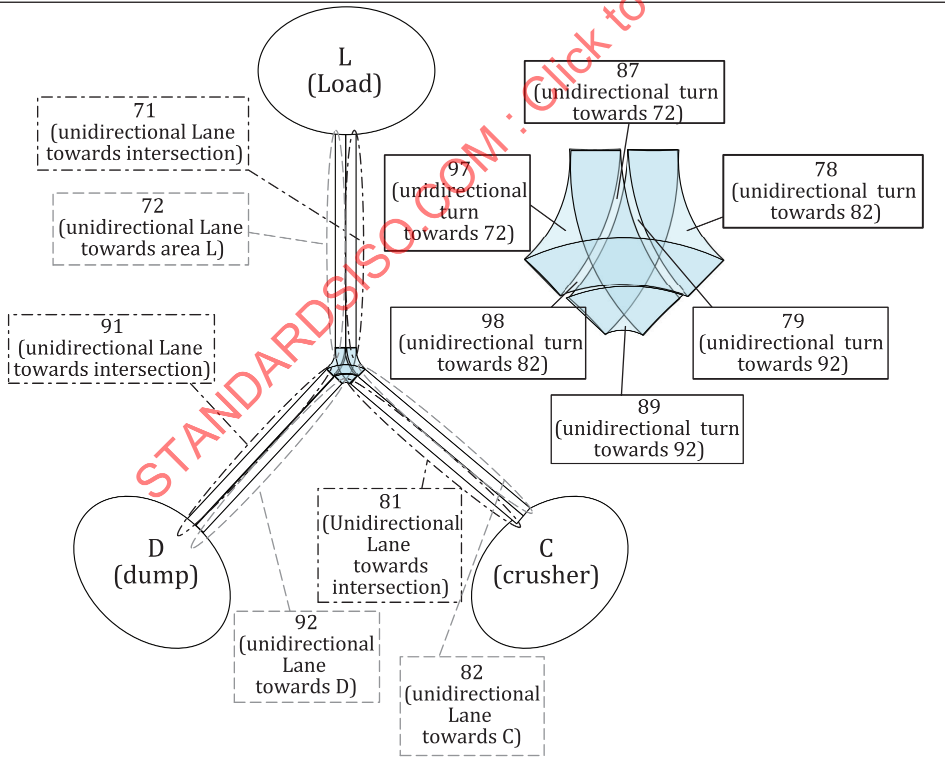


Figure 5 — Example of lane level modelling

Dumping Area (D) connected to — (91) Crusher Area (C) connected to — (81) Loading Area (L) connected to — (71) Segment (92) connected to — (D) Segment (72) connected to — (L) Segment (82) connected to — (C) Segment (91) connected to — (72) — (82) Segment (71) connected to — (82) — (92) Segment (81) connected to — (72) — (92)	
Dispatching Instructions from Loading to Dump	
— (L) — (71) — (92) — (D)	—
 L (Load) 87 (unidirectional turn towards 72) 71 (unidirectional Lane towards intersection) 97 (unidirectional turn towards 72) 78 (unidirectional turn towards 82) 72 (unidirectional Lane towards area L) 91 (unidirectional Lane towards intersection) 98 (unidirectional turn towards 82) 79 (unidirectional turn towards 92) 89 (unidirectional turn towards 92) 81 (Unidirectional Lane towards intersection) D (dump) C (crusher) 92 (unidirectional Lane towards D) 82 (unidirectional Lane towards C)	— (L) — (71) — (79) — (92) — (D)
Figure 6 — Example of intersection level modelling	Dispatching instructions loading to dump

Dumping Area (D) connected to — (91) Crusher Area (C) connected to — (81) Loading Area (L) connected to — (71) Segment (92) connected to — (D) Segment (72) connected to — (L) Segment (82) connected to — (C) Segment (91) connected to — (97) — (98) Segment (71) connected to — (78) — (79) Segment (81) connected to — (87) — (89) Segment (78) connected to — (82) Segment (79) connected to — (92) Segment (87) connected to — (72) Segment (89) connected to — (92) Segment (98) connected to — (82) Segment (97) connected to — (72)	

Table 42 — Example of a road and area connectivity message

Connectivity message

```
{
  "Protocol":"ISO23725",
  "Version": 1,
  "Timestamp": "2018-10-31T09:30:10.435Z",

"MapConnectivityV1": [
  {
  "WayId": 750321441,
  "WayType": "Road",
  "RoadProperty": { "Length": 12.5 },
  "Connection": [ {"WayId": 750321442} ]
  },
  {
  "WayId": 750321442,
  "WayType": "Area",
  "AreaProperty":{"DefaultTask": "Load", "Name":"skookum"},
  "Connection": [ {"WayId": 750321440} ]
  },
  {
  "WayId": 750321440,
  "WayType": "Road",
  "RoadProperty":{ "Length": 10.2 },
  "Connection": [ {"WayId": 750321439} ]
  },
  {
  "WayId": 750321436,
  "WayType": "Road",
  "RoadProperty":{ "Length": 65.9  },
  "Connection": [ {"WayId": 750321439} ]
  },



  {
  "WayId": 750321439,
  "WayType": "Road",
  "RoadProperty":{ "Length": 52.5  },
  "Connection": [
    {"WayId": 750321350},
    {"WayId": 750321398}
    ]
  },
  {
  "WayId": 754062815,
  "WayType": "Area",
  "AreaProperty":{"DefaultTask": "Offload"},
  "Connection": [ {"WayId": 754062809} ]
  },
```

Table 42 *(continued)*

```
  {
  "WayId": 754062805,
  "WayType": "Road",
  "RoadProperty":{ "Length": 24.3  },
  "Connection": [ {"WayId": 754062815} ]
  },
  {
  "WayId": 754062809,
  "WayType": "Road",
  "RoadProperty":{ "Length": 79.6  },
  "Connection": [
    {"WayId": 754062806},
    { "WayId": 754062808}
    ]
  },
  {
  "WayId": 754062812,
  "WayType": "Area",
  "AreaProperty":{"DefaultTask": "Offload", "Name":"W.C."},
  "Connection": [ {"WayId": 754062810} ]
  },
  {
  "WayId": 754062811,
  "WayType": "Road",
  "RoadProperty":{"Length": 58.2  },
  "Connection": [ {"WayId": 754062812} ]
  }
  ]
}
```

10 Map service

The map service shall use a session and transport layer as defined in 4.1. The map service shall use a separate communication channel. The map service shall not use this channel to multiplex messages destined for the AHS-FMS communication. The map service shall serve a single map; the service does not support distinctly addressable independent maps.

10.1 Map object concepts

a) A map shall be composed of roads and areas.

b) A road shall be modelled as a line where the start and end points are different.

c) An area shall be modelled as a line where the start and end points are the same.

d) A line shall be modelled as a way element composed of an ordered list of points.

e) A point shall be modelled as a node element.

10.2 Map object identification number rules and restrictions

10.2.1 Restrictions on object identification numbers

All map objects shall have a unique identification number within their class.

The identification number of map objects shall be immutable throughout the object's life cycle, but their attributes may change in value. A unique identification number is created when the object is created and only discarded when the object is deleted, see Table 43.

Table 43 — Specific map, object identification rules and restrictions

Object	Restrictions
node id	0 is invalid. Each node shall have a unique ID (amongst all Nodes) in the map held on the Map Server.
way id	0 is invalid. Each Way shall have a unique ID (amongst all Ways) in the map held on the Map Server.
user id	0 is invalid. Each user shall have a unique ID (amongst all users) in the Map Server.

10.2.2 Identification numbers rules for ways and nodes

Identification numbers of ways shall remain immutable after a Map update (change set of edits) when:

- The map features they describe are not modified at all, or
- If node references inside a way were modified (moved, added or deleted) but the feature still describes the same object (lane, road or shape), or
- If tags were changed, (added or removed) to a map feature.

Consequently, the list of way ids in the map shall change when:

- A new feature (lane, road or area) is added to the Map. In other words, a new "way id" is added to the "way id" list and the ChangeSetWay is incremented.
- An element is deleted from the Map. In other words, "way id" is removed from the "way id" set and the ChangeSetWay is incremented.

Identification numbers of nodes shall remain immutable after a Map update (change set of edits) when

- nodes were not modified at all, or
- the Lat or Lon attributes were modified, or
- tags were changed, added or removed.

Consequently, the list of nodes ids in the map shall change when:

- A new node is added
- A node is deleted

10.3 Map format

The map document format is based on the Open Street Map format[1].

10.3.1 Map document encoding

The map document shall be contained in an XML document (ISO 8879).

The XML document shall not be encapsulated in a JSON object.

A map document shall start with this XML prolog[2]:

```
<?xml version='1.0' encoding='UTF-8'?>
```

[1] Public documentation location: https://wiki.openstreetmap.org/wiki/OSM_XML.

[2] Ref: https://www.w3schools.com/XML/xml_syntax.asp

A map document shall contain exactly one ‘osm element’ serialized in an XML format.

```
<osm version='0.6' generator='JOSM' sequence='1'>
</osm>
```

The ‘osm element’ shall solely contain these attributes:

- A ‘version attribute’ that shall be set to 0.6
- A ‘generator attribute’ naming the software that generated the map
- A ‘sequence attribute’ that is the sequence number of the current map chunk as large maps are segmented into multiple XML documents.

The ‘osm element’ shall aggregate these elements:

- ‘node element’
- ‘way element’

10.3.2 Splitting a large map into chunks

Each map chunk shall respect the map document encoding as defined by 10.3.1.

A map document shall have a minimum of one chunk.

‘XML element’ integrity shall be maintain when splitting the map into chunks to preserve the validity of XML. Each chunk shall be grammatically valid when parsed by a standard XML parser.

Each chunk shall be bound by these upper limits:

- maximum of 10 000 elements per chunk (combined points, roads and areas);
- maximum of 2 000 nodes per way (maximum number of points per road or area).

Each map chunk shall have a sequence number encoded in the osm element as a sequence attribute.

The first chunk shall bare the sequence number ‘1’. Each subsequent ordered chunk shall have their sequence number equal to the previous chunk+1.

The map service shall manage the possibility of a concurrent new map changesets so that the map chunk transfer remains consistent with the latest Map summary sent to the client application.

10.3.3 Point – node element

A point shall be encoded as a ‘node element’. The attributes of a ‘node element’ shall conform to Table 44. Below is the minimum attribute set of a node:

```
<node  id='xxx'  lat='yy.yyyyyy'  lon='zzz.zzzzzzz'  />
```

Table 44 — Attributes of a ‘node element’

Attribute	Requirement Level	Format	Description
id	shall	64 bit integer (not equal to zero)	Node ids are unique within the map service node numbering space. Map authoring tools should use negative node ids when saving work in a local file that is not merged into the map service. Negative node ids should denote that these nodes have not been merged with the Map service yet. The Map service should only serve positive node ids.
lat	shall	decimal number ≥ −90,000 000 0 and ≤ 90,000 000 0 (with 7 decimal places)	Latitude coordinate in degrees (North of equator is positive) using the standard WGS84 coordinate system.
lon	shall	decimal number ≥ −180,000 000 0 and ≤ 180,000 000 0 (with 7 decimal places)	Longitude coordinate in degrees (East of Greenwich is positive) using the standard WGS84 coordinate system.
ele	should	decimal number in meters	metres above mean sea level as defined by the EGM96 geoid model.
timestamp	should	Timestamp	UTC Date Time stamp as per ISO 8601 (all parts)
changeset	should	64 bit integer	The id of the batch transaction done with the map service once a user posts a changeset.
version	should	64 bit integer	The edit version of the object. Newly created objects start at version 1 and the value is incremented by the server when a client uploads a new version of the object.
uid	should	64 bit integer	User Id as defined by the Map service that last edited and posted the changeset. uid management is out of scope of this document.
user	may	string	The human readable and printable name of the user that last edited and posted the changeset
visible	should	‘true’ || ‘false’	If it should be drawn on the map

10.3.4 Line – way element

A line shall be encoded as a ‘way element’. The attributes of ‘way element’ shall conform to Table 45. The ‘way element’ shall aggregate multiple ‘nd element’. The attributes of ‘nd element’ shall be a ref attribute that is equal to a previously defined point identification: <node id=’yyy’ … />.

```
<way  id='xxx'  >
  <nd ref='yyy' />
…
  <nd ref='zzz' />
</way>
```

Table 45 — Attributes of a ‘`way` element’

Attribute	Requirement Level	Format	Description
id	shall	64 bit integer	way ids are unique within the map service node numbering space. Map authoring tools should use negative way ids when saving work in a local file that is not merged into the map service. Negative way ids should denote that these ways have not been merged with the Map service yet. The Map service should only serve positive way ids.
timestamp	should	Timestamp	UTC Date Time stamp as per ISO 8601 (all parts)
changeset	should	64 bit integer	The id of the batch transaction done with the map service once a user posts a changeset.
version	should	64 bit integer	The edit version of the object. Newly created objects start at version 1 and the value is incremented by the server when a client uploads a new version of the object.
uid	should	64 bit integer (not equal to zero)	User Id as defined by the Map service that last edited and posted the changeset. uid management is out of scope of this document.
user	may	string	The human readable and printable name of the user that last edited and posted the changeset
visible	should	‘true’ || ‘false’	If it should be drawn on the map

10.3.5 Road

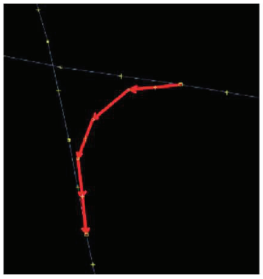
A road shall be modelled as a line, defined by an ordered sequence of points, where the first point and the last point shall be different. A road shall have a minimum of 2 points and a maximum of 2 000 points. A road shall not cross or connect to itself. Different roads can cross each other.

Roads describe the path machines should follow in normal driving conditions. How autonomous machines track their position in relation to the lines is Autonomy System specific.

10.3.6 Lane

Lanes shall be modelled as a unidirectional road. Lanes should have a tag with a oneway = yes attribute, see Table 46..

Table 46 — Road or lane example inside an XML document



A Line object with referenced nodes representing a lane

```
<?xml version='1.0' encoding='UTF-8'?>
<osm version='0.6' generator='JOSM'>
<node id='7015834148' timestamp='2019-11-28T01:01:05Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77656912' lat='-23.20035341832' lon='118.78111995623'
/>
<node id='7015834149' timestamp='2019-11-28T01:01:05Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77656912' lat='-23.20030750198' lon='118.78113940224'
/>
<node id='7015834150' timestamp='2019-11-28T01:01:05Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77656912' lat='-23.20027206318' lon='118.7811863409' />
<node id='7015834151' timestamp='2019-11-28T01:01:05Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77656912' lat='-23.20026559174' lon='118.78125473723'
/>

<way id='750321398' timestamp='2019-11-28T17:22:58Z' uid='10092170' user='Martin Politick'
visible='true' version='2' changeset='77694095'>
    <nd ref='7015834151' />
    <nd ref='7015834150' />
    <nd ref='7015834149' />
    <nd ref='7015834148' />
    <nd ref='7015834272' />
    <tag k='highway' v='service' />
    <tag k='oneway' v='yes' />
  </way>
</osm>
```

10.3.7 Area

An area shall be modelled as a line, defined by an ordered sequence of points, where the first point and the last point shall have the same node id. An area shall have a minimum of 3 points and a maximum of 2 000 points. The line defining the area perimeter shall not cross itself. Different areas can overlap each other, see Table 47.

Areas describe a surface on earth where other objects can be “inside of” or “outside of” to accomplish a function. The direction of rotation (clockwise or counter clockwise) of the ordered nodes has no intrinsic meaning.

Purpose:

Areas are used for multiple functions in a map:

a) Define an open drivable surface where machines can create their own free form planning path with minimal risks of obstacles to reach their destination in an area. In mining, this is mainly used for locations that change shape as the material is mined:

   1) the active mining face where trucks are loaded;

   2) the waste dump where waste is dumped over the edge;

   3) on a paddock to create lifts.

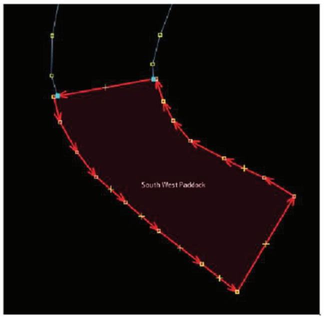
but also more permanent areas like

- a parking, and
- a fuel island.

b) Area(s) where autonomy is restricted to; called "Autonomous Operating Zone" or "AOZ". This defines one or more LARGE polygons where autonomous equipment are allowed to operate in autonomous mode while they are inside an AOZ polygon. The AOZ will overlap all the map objects that can be used by autonomous machines.

c) Areas inside the AOZ where autonomous machines are excluded from. An autonomy excluded area can partially or completely overlap with an autonomous area.

Table 47 — Example of an area object inside an XML document



Area object inside XML document (with referenced nodes)

```
<?xml version='1.0' encoding='UTF-8'?>
<osm version='0.6' generator='JOSM'>
<node id='7033590282' timestamp='2019-12-04T15:37:42Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77949246' lat='-23.20153521943' lon='118.77237528428'
/>
<node id='7033590283' timestamp='2019-12-04T15:37:42Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77949246' lat='-23.20159438626' lon='118.7724088119' />
<node id='7033590284' timestamp='2019-12-04T15:37:42Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77949246' lat='-23.20165478569' lon='118.7724678205' />
<node id='7033590285' timestamp='2019-12-04T15:37:42Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77949246' lat='-23.20171148718' lon='118.77258047327'
/>
<node id='7033590286' timestamp='2019-12-04T15:37:42Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77949246' lat='-23.20176818864' lon='118.77271726593'
/>
<node id='7033590287' timestamp='2019-12-04T15:37:42Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77949246' lat='-23.20182489008' lon='118.77281784877'
/>
<node id='7033590288' timestamp='2019-12-04T15:37:42Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77949246' lat='-23.20212195592' lon='118.77262472972'
/>
<node id='7033590289' timestamp='2019-12-04T15:37:42Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77949246' lat='-23.20203690395' lon='118.77250537142'
/>
<node id='7033590290' timestamp='2019-12-04T15:37:42Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77949246' lat='-23.20193213031' lon='118.77236053281'
/>
```

Table 47 *(continued)*

```
<node id='7033590291' timestamp='2019-12-04T15:37:42Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77949246' lat='-23.20184399661' lon='118.77224855058'
/>
<node id='7033590292' timestamp='2019-12-04T15:37:42Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77949246' lat='-23.20176572398' lon='118.77215064995'
/>
<node id='7033590293' timestamp='2019-12-04T15:37:42Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77949246' lat='-23.20169238186' lon='118.77208426528'
/>
<node id='7033590294' timestamp='2019-12-04T15:37:42Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77949246' lat='-23.20159931744' lon='118.77202726834'
/>
<node id='7033590295' timestamp='2019-12-04T15:37:42Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77949246' lat='-23.2015191957' lon='118.77200648122' />
<node id='7033590296' timestamp='2019-12-04T15:37:42Z' uid='10092170' user='Martin Politick'
visible='true' version='1' changeset='77949246' lat='-23.20146434309' lon='118.77235382728'
/>

<way id='752673531' timestamp='2019-12-09T18:51:56Z' uid='10092170' user='Martin Politick'
visible='true' version='2' changeset='78167348'>
    <nd ref='7033590296' />
    <nd ref='7033590295' />
    <nd ref='7033590294' />
    <nd ref='7033590293' />
    <nd ref='7033590292' />
    <nd ref='7033590291' />
    <nd ref='7033590290' />
    <nd ref='7033590289' />
    <nd ref='7033590288' />
    <nd ref='7033590287' />
    <nd ref='7033590286' />
    <nd ref='7033590285' />
    <nd ref='7033590284' />
    <nd ref='7033590283' />
    <nd ref='7033590282' />
    <nd ref='7033590296' />
    <tag k='autonomy' v='exclusive' />
    <tag k='landuse' v='quarry' />
    <tag k='name' v='South West Paddock' />
    <tag k='default_task' v='offload' />
  </way>
</osm>
```

10.4 Adding context to objects

Map objects should be annotated with as many tag elements as available to enhance the context of their purpose.

The map format defines an encoding method using tag elements to enhance the context of map objects (points, roads and areas). These optional tag elements bring more context to objects so client software consuming the Map can extract important information for their particular roles.

10.4.1 Tag Element Attribute Encoding

The map server shall use an XML tag element with 2 attributes, k and v, inside the scope of an object to bring more context to the map object,

```
<tag k='xxx' v='yyy' />
```

Where xxx and yyy are the attribute value of the key ‘k’ and its associated value ‘v’. There shall be a single ‘k’ and a single ‘v’ per tag. Multiple attributes are encoded by adding multiple tags in a single map object scope, see Table 48..

Table 48 — Example of multiple tags contextualizing an area

```
  <way id='750321396' timestamp='2020-03-17T21:23:51Z' uid='10092170' user='Martin Politick'
visible='true' version='3' changeset='82318740'>
    <nd ref='7015834140' />
    <nd ref='7015834219' />
    <nd ref='7015834441' />
    <nd ref='7015834115' />
    <nd ref='7015834114' />
    <nd ref='7015834113' />
    <nd ref='7015834140' />
    <tag k='amenity' v='parking' />
    <tag k='name' v='East South Pit Parking' />
    <tag k='default_task' v='park' />
  </way>
```

10.4.2 Standard tags

Table 49 — Map service supporting these tags

Tag Key	Tag values
name	“Any human readable string”
default_task	{load, offload, supply, park, wait, drive, fuel}
autonomy	{excluded, exclusive, mixed}
autonomy:area	{AOZ, exclusion, obstacle}
oneway	{yes, no, reversible, alternating}
highway	{service, track, escape, construction}
amenity	{parking, fuel}
landuse	{quarry, construction}

10.4.3 Tag with Attribute *k* = ’name’

Purpose:

The name is the primary identifier people use to reference map objects when talking together. Encoding names to important objects is standard practice. All roads and areas named in the map authoring tool should be exported to the map document. A name does not have to be unique within the map. It is common to give the same name to multiple objects, see Table 50.

Table 50 — Name usage on map objects

Object type	Description
node	Supported. A point of interest identifying something meaningful to people. The text will carry the meaning of the point of interest.
way as road	Supported. Important roads and road segments are often given names so people can reference them by name.
way as area	Supported. Important areas are often given names so people can reference them.

10.4.4 Tag with Attribute *k* = 'default task'

Purpose:

The primarily usage of the default_task is when a third-party software consumes the map, then the software can identify how the different area are designed to be utilized from a mining perspective and enable the appropriate behaviours and attributes for these individual areas. This tag is a cue and does not enforce or restrict the tasks that can be assigned to a machine in this area, see Tables 51 and 52.

Table 51 — Default_task usage on map objects

Object type	Description
node	Not Supported.
way as road	Not Supported.
way as area	Supported. It defines the primary mining purpose of an area

Table 52 — Default_task key values

Key value	Description
load	An area where the machine will primarily pickup a load.
offload	An area where the machine will primarily offload their load.
supply	An area where the machine will can pickup or offload a load.
park	An area where the machine will primarily park and shutdown their prime mover for an extended period.
wait	An area where the machine will primarily stop and wait for further instructions. The main difference with park is that the machine in on standby and it is not expected to shutdown their prime mover.
drive	An area where the machine will primarily drive through to reach another road or area. The difference between this case and road is that a drive area does not have an explicit path to follow and the machine will have to plan a freeform path.
fuel	An area where the machine will primarily re-fuel. Either a fuel island or where a fuel truck is present.

10.4.5 Tag with Attribute *k* = 'autonomy'

Purpose:

The autonomy attribute is used by third-party software to identify which areas are designed to be used by autonomous machines and which areas they are excluded from. When the map service is serving the entire mine, both the autonomous and the manned areas, then this attribute allows the client applications to distinguish where autonomous machines are allowed to be dispatched without an exception, see Tables 53 and 54.

Table 53 — Autonomy usage on map objects

Object type	Description
node	Not Supported.
way as road	Supported. The autonomy attribute can be associated with a road to specify if autonomous machines are allowed on this road.
way as area	Supported. The autonomy attribute can be associated with an area to specify if autonomous machines are allowed to plan paths in this area.

Table 54 — Autonomy key values

Key value	Description
excluded	Autonomous machines are NOT allowed to use this map object. Only vehicles operated by people are allowed to use this map object.
exclusive	ONLY Autonomous machines are allowed to use this map object. Hence vehicles operated by people are not allowed to use this map object.
Mixed	All operation types (autonomous and manned) are allowed to use this object.

10.4.6 Tag with Attribute *k* = ’autonomy:area’

10.4.6.1 Purpose

The autonomy:area attribute is used by third-party software to identify perimeters that are solely designed to express if autonomous machines are allowed or disallowed inside their perimeter, see Tables 55 and 56.

The autonomy:area are typically overlapped on drivable areas, they are not meant to be drivable surfaces.

Table 55 — Autonomy: area usage on map objects

Object type	Description
node	Not Supported.
way as road	Not Supported.
way as area	Supported. The autonomy: area attribute is used to define if the perimeter defined is inclusionary or exclusionary of autonomous machines.

Table 56 — Autonomy: area key values

AOZ	Autonomous Operating Zone. This perimeter(s) encompasses all roads and areas where autonomous machines are allowed to operate autonomously. Machines can’t be autonomous outside of an AOZ perimeter. The use of AOZ is optional, but when used, the AOZ shall be a permanent object.
exclusion	Machines running autonomously shall not touch or enter an exclusion perimeter. These exclusion perimeters have a lifecycle counted in hours or days.
obstacle	Machines running autonomously shall not touch or enter the perimeter of an obstacle. Obstacles are temporary and have a lifecycle counted in minutes to a few hours.

Group will need to debate if exclusions and obstacles should have their own scope inside the map service. The reason is that exclusions and obstacles are temporary in nature, almost a real-time component. Where the rest of the map, are not. So far, the objects represented in the mapping scope are quasi-permanent as they need to be built.

10.4.7 Tag with Attribute *k* = ’oneway’

10.4.7.1 Purpose

The oneway attribute is used by third-party software to identify lanes or one way roads. Lanes are modelled as one-way roads. The order of points in the line defines the direction of the one-way. The first point is the entrance and the last point is the exit, see Tables 57 and 58.

This tag shall not be used for dispatch decision purposes.

Table 57 — Oneway usage on map objects

Object type	Description
node	Not Supported.
way as road	Supported. Specifies if the road is to be used in a single direction.
way as area	Not Supported.

Table 58 — Oneway key values

yes	Indicates that traffic can flow in a single direction, from the first to the last point.
no	Indicates that the traffic is bidirectional. Roads are interpreted as bidirectional when without a oneway tag. Omitting the oneway tag is equivalent to <tag k='oneway' v='no'/>
reversible	Roads that alternate between different directions regularly but infrequently. E.g. a lane that changes direction during rush hour, twice a day.
alternating	Roads that alternate between different directions regularly and often or continuously. E.g. crossing a bridge with a single machine width, typically with lights alternating the flow of machines as needed to share the resource.

10.5 Map building

10.5.1 Connecting map objects

For 2 map objects to be considered connected, the 2 objects shall fulfil one of these conditions:

a) Two roads share a single common point.

b) One road and one area share a single common point.

c) Two areas share 2 common points.

Consequently, two map objects that share the same spaces but don't share a point are not connected. They could possibly be at two different elevations, like a viaduct or a bench, or they may not be accessible because of a fence or administrative rule.

10.5.2 Building a purely structured road model

A purely structured road model allows unambiguous routing and road segment ingress/egress constraints. A road segment is a portion of the road where when a machine enters at one end, it can only exit at the other end. Meaning that no junction, merge points or intersections exists along that road segment. This model is fully explicit and no interpretation via external rules is required to interpret the road network.

The map service shall serve purely structured maps.

10.5.3 Purely structured rules

A purely structured map shall conform to all of these conditions:

a) Road segments shall have 2 ends, the first and last point of the line defining the road segment.

b) Areas shall strictly have 0 end.

c) Only the ends of a road segment shall be connected to other road segments.

d) A road segment shall not connect to itself.

e) Machines shall only ingress or egress a road segment at an end.

f) Machines shall ingress or egress an area from the end of a connected road segment.

g) Machines shall ingress or egress to and from 2 connected areas.

h) A road segment can be connected to an area by sharing any point along its line.

i) Roads and areas are considered connected if the share a common point ID. Features that don't share a point ID shall be considered disconnected.

10.6 Map service messaging

All map service messages shall include a message header as defined in Table 3.

10.6.1 Map service identification

The map service identification message shall include the structure defined in Table 60. A Map service identification message shall be immediately sent each time a new client connects to the service.

Table 59 specifies which actor is allowed to send the message and when this message is expected.

Table 59 — Messages initiator and trigger

	MAP to FMS	FMS to MAP
Message direction allowed	Yes	No
Trigger: at connection	Yes	No
Trigger: runtime change	No	No

10.6.2 Map Service Identification V1

Table 60 — Map Service Identification V1

Member	Req. Level	Type	Description
MapServiceV1	shall	Object	Identifies this structure as a Map Service identification.
API	shall	integer	The API level supported by the Map service. API shall be set to 1

An example of a valid map service identification message is given in Table 61.

Table 61 — Map service identification message

The Map service identifies itself and the API level it supports.

```
{
   "Protocol":"ISO23725",
   "Version": 1,
   "Timestamp": "2020-07-24T09:30:10.831Z",

   "MapServiceV1": {
   "API": 1
   }
}
```

10.6.3 Map summary

The map summary message shall include the structure defined in Table 62. A Map Summary message shall be sent once:

— a new connection to the server is established by a client.

— a new change set is posted to the map

— a connected client requests a summary message as defined in Table 66

— just before a map document is sent

The purpose of the message is to inform connected clients about internal properties of the current map without sending the entire map.

Table 62 — Map summary messaging initiator and trigger

Trigger	MAP->FMS	FMS->MAP
At Connection	Yes	No
On Map change	Yes	No
On client request	Yes	No

Table 63 — Map summary message members

Member	Req. Level	Type	Description
MapSummaryV1	shall	Object	Identifies this structure as a Map SummaryV1
Timestamp	shall	ISO 8601 (all parts)	The UTC time the map was last modified
ChangeSetMap	shall	Integer(64)	The serial number of the post (set of changes) to the map. Each map commit will increment the map set number by 1.
ChangeSetWay	shall	Integer(64)	The serial number of changes to the Ways only. Ways are deemed changed in the map if WayIds are removed or added and if tags to a way are modified, added or removed.
uid	should	Integer(64)	A User ID that modified it (this uid is specific to the Map service and may not be resolved by other open autonomy services)
NodeCount	shall	Integer(64)	The number of <node /> objects (Points)
WayCount	shall	Integer(64)	The number of <way /> objects (Roads or Area)
DocumentCount	shall	Integer(64)	The number of XML documents (map chunks) that will be sent when a full update is sent.

ChangeSet: When the service starts, the implementor can re-start the ChangeSet number to 1 or persist the last Changeset before the shutdown, either implementation is valid. As for the client application, if it receives a changeset number that is less than the one it has in its cache, then something significant happened with the Map service (maybe the map service rebooted or an older map was restored), so the client should discard its current autonomous map cache and request a full download.

Ther ChangeSetWay is used to allow the client application to know when the road structure has changed, in opposition to just the road geometry. Entity nodes (its geometry) may change relatively often over time, but that the number of roads and areas, these should remain relatively constant during a shift.

Table 64 — Map summary messaging example

The Map service returns the Map metadata

```
{
  "Protocol":"ISO23725",
  "Version": 1,
  "Timestamp": "2020-07-24T09:30:10.948Z ",

  "MapSummaryV1": {
    "Timestamp": "2020-07-22T21:35:27.226Z ",
    "ChangeSetMap":82318740,
    "ChangeSetWay":598902,
    "uid": 10092170,
    "NodeCount": 36541,
    "WayCount": 3853,
    "DocumentCount": 5
  }
}
```

10.6.4 Map command – get summary

The Get Summary message shall include the structure defined in Table 66. A get summary message may be sent by a client software as the client software requires the map summary information from the server.

Table 65 — Map command – get summary initiator and trigger

Trigger	MAP->Client	Client->MAP
At connection	No	No
Runtime polling	No	Yes

Table 66 — Map command – get summary message members

Member	Req. Level	Type	Description
MapCommandV1	shall	Object	Identifies this structure as a Map CommandV1
Method	shall	string	The method value shall be set to “GET”
Scope	shall	string	The scope value shall be set to “/summary”

Table 67 — Map command - get summary example

Client request the metadata of the map from the /Map service

```
{
   "Protocol":"ISO23725",
   "Version": 1,
   "Timestamp": "2020-07-24T09:30:10.831Z ",

   "MapCommandV1":
   {
      "Method":"GET",
      "Scope": "/summary"
   }
}
```

10.6.5 Map command – get all

The Get All message shall include the structure defined in Table 69. A get all message should be sent by a client software as the client software requires the map content from the server. The purpose of this command is to trigger a transmission of the entire Map from the Map server.

Table 68 — Map command – get all initiator and trigger

Trigger	MAP->FMS	FMS->MAP
At connection	No	No
Runtime on demand	No	Yes

Table 69 — Map command – get all message members

Member	Req. Level	Type	Description
MapCommandV1	shall	Object	Identifies this structure as a Map CommandV1
Method	shall	string	The method value shall be set to “GET”
Scope	shall	string	The scope value shall be set to “/”

Table 70 — Map command – get all example

Client request the whole Map from the Map service

```
{
  "Protocol":"ISO23725",
  "Version": 1,
  "Timestamp": "2018-10-31T09:30:10.435Z",

  "MapCommandV1":
  {
    "Method":"GET",
    "Scope": "/"
  }
}
```

10.6.6 Map document message

Upon receiving a get all message (see Table 69) from a client application, the map server shall transmit the entire map information using one, or multiple chunks.

If the map is empty, the server shall return this XML document with one empty OSM element.

Table 71 — XML document with an empty OSM element

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" generator="AHS server" sequence=”1” />
```

Table 72 — Map document message – simple single lane example

The Map service returns the XML document of the map. This simplistic example is a single lane.

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" generator="AHS Authoring" sequence="1">
 <node id="7015812542" changeset="77656912" timestamp="2019-11-28T01:01:04Z" uid="10092170"
lat="-23.1850242" lon="118.7917757"/>
 <node id="7015812543" changeset="77656912" timestamp="2019-11-28T01:01:04Z" uid="10092170"
lat="-23.1828840" lon="118.8020003"/>
 <node id="7015812544" changeset="77656912" timestamp="2019-11-28T01:01:04Z" uid="10092170"
lat="-23.1806847" lon="118.8050794"/>
 <node id="7015812545" changeset="77656912" timestamp="2019-11-28T01:01:04Z" uid="10092170"
lat="-23.1765719" lon="118.8010454"/>
 <node id="7015812546" changeset="77656912" timestamp="2019-11-28T01:01:04Z" uid="10092170"
lat="-23.1703876" lon="118.7992537"/>
 <way id="754062802"  changeset="78167348" timestamp="2019-12-09T18:51:55Z" uid="10092170">
  <nd ref="7015812542"/>
  <nd ref="7015812543"/>
  <nd ref="7015812544"/>
  <nd ref="7015812545"/>
  <nd ref="7015812546"/>
  <tag k="highway" v="service"/>
 </way>
</osm>
```

11 Interface diagnostic

A service diagnostic stream should be implemented.

If implemented, the service diagnostic stream shall be separate from the main service interface stream.

11.1 Interface diagnostic information

11.1.1 Diagnostic measurement information

Upon connection, this diagnostic information will be available:

- protocol version of the main interface;
- message Level Version of the main interface;
- number of active connections on the main interface;
- number of active connections on the diagnostic interface;
- byte count In/Out for each connection on the main interface;
- message count In/Out for each connection on the main interface;
- network load;
- CPU load;
- memory load;
- public diagnostic.

This will be available over a web socket wss://protocol.

11.1.2 Diagnostic message information

Support of Verbosity level on the Service is optional but should be supported. The Client connected shall be able to request a change in verbosity level and the Service may comply.

Diagnostic measurements defined in 11.1.1 shall always be transmitted, regardless of verbosity level.

- Level 0 - Silent - No Diagnostic Message.
- Level 10 - Fatal - Critical System Error code. This means that the server will require a restart.
- Level 20 - Error - Errors including mis-configurations and importing disabling part of the functionality.
- Level 30 - Warning - Degraded performances and user errors.
- Level 40 - Info - Normal logging level, with All that's going on and normal transactions are printed out, usable by an end user for monitoring. All Raw incoming and outgoing Messages should be printed at this level.
- level 50 - Debug - Trouble shooting something, aimed more at Tier 2 and 3 support from OEM
- level 60 - Trace - Engineering level tracing, this level may impact the server performance.
- level 70 - Ludicrous - The descriptions summarize it well, should not be in a release version.

The interface would periodically (~15 s) output a diagnostic measurement message as seen in the following example:

```
{
 "Iso23725DiagnosticMeasurementV1": {

  "ProtocolVersion": 1,
  "ReleaseLevelVersion": 2,
  "MainConnection": 3,
  "DiagnosticConnection": 4,
  "MainByteIn":  [ 4456356256, 603917, 185356203 ],
  "MainByteOut": [ 584646735, 48291, 9563021 ],
  "MainMessageIn": [ 5635625, 4626, 85672 ],
  "MainMessageOut": [ 8464673, 6784, 78521],
  "NetworkLoad": 57,
  "CPULoad":  38,
  "MemoryLoad": 72
 }
}



---A client is requesting a more verbose level to the service
{
  "Iso23725DiagnosticMessageLevelV1" : {
  "Level" : 50
  }
}



--- A Debug level message is sent from the service and output on the diagnostic stream
websocket
{
  "Iso23725DiagnosticMessageV1" : {
  "Timestamp": "2018-10-31T09:30:10.43511Z",
  "Code": 67378,
  "Level": 50,
  "Data": "The Orange Heap thing is on line 455 of MyThingAMaJig.cpp"
  }
}



----- an incoming message from AHS just got received from the FMS service:
{ "Iso23725DiagnosticMessageV1":
  {
  "Timestamp": "2018-10-31T09:30:11.497Z",
  "Code": 3,
  "Level": 40,
  "Data": "Incoming Message from 10.0.250.12:58478 AHS=\"2248d535-3daf-4a86-b1e1-
4951a22beec6\"::{
```

```
\"EquipmentId\":\"66c89998-8e06-5d11-9e25-4a517e99ff92\",
\"Protocol\": \"ISO23725\",\"TimeStamp\": \"2020-06-10T13:47:44.8271571Z\",
\"MachinePositionV1\": {\"Elevation\": 27.12450457137311 ,\"Heading\": 359.98077333877325
,\"HeadingAccuracy\": 100.0,\"HeightAccuracy\": 100.0,\"Latitude\": 41.74923997529961
,\"LatitudeAccuracy\": 100.0,\"Longitude\": -112.02557557709049,\"LongitudeAccuracy\":
100.0,\"Speed\": 0},\"Version\": 1}"
  }
}
```

12 Haul truck production

12.1 Message content

Haul truck production messages shall

- have an `EquipmentId` attribute in their header, and
- contain a MachinePosition object in their body, as defined in Table 20.

12.2 Production cycle message

Haul Truck production cycle information to determine state shall be sent in one of 2 ways

a) As already onboard determined haul truck production states as described in 12.4

b) As a sensor information stream that can be processed offboard by the FMS to produce haul truck production states. As described in 12.5

The onboard processing is the preferred method and the most accurate.

12.3 Production cycle definition

A haul truck production cycle shall starts with an empty tray waiting for dispatching instructions.

- After receiving instructions, the truck will be “Driving” with an empty tray while moving towards its destination,
- Once it reaches its destination and ready to receive a payload then it will go in “Loading” by a shovel, loader, excavator, etc.,
- then it will receive new instructions where to bring the payload and start “Hauling” with payload in the tray,
- then eventually “Dumping” emptying the tray and start a new cycle.

12.3.1 Definition of a haul truck cycle

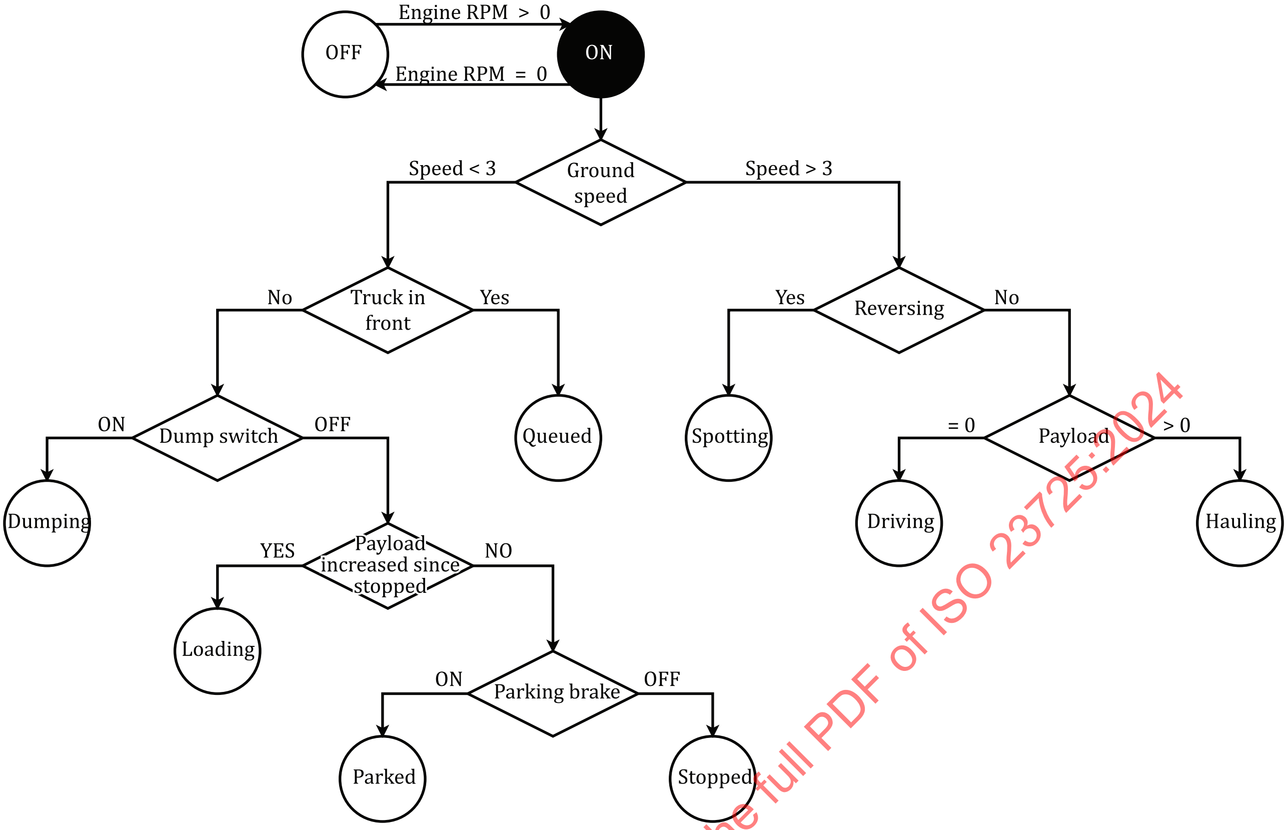


Figure 7 — Haul truck cycle

The production cycle will encounter other states than these 4 during production, but in essence a production cycle is defined by: the hauler needs to “go somewhere”, “get a load of something”, “go somewhere else” and then “dump the load”. Emptying the tray shall complete a production cycle.

12.3.2 Proposed state decision tree

The AHS should implement the decision tree in Figure 8 or its own variation to keep track of the truck production state, see Table 73. The AHS provider shall submit documentation explaining what are the conditions to reach each production state. Note that this is not a state machine so the current production state can transition to any other state in the tree; also there are variations on the implementation that can reach the same intended result of determining the haul truck production state.

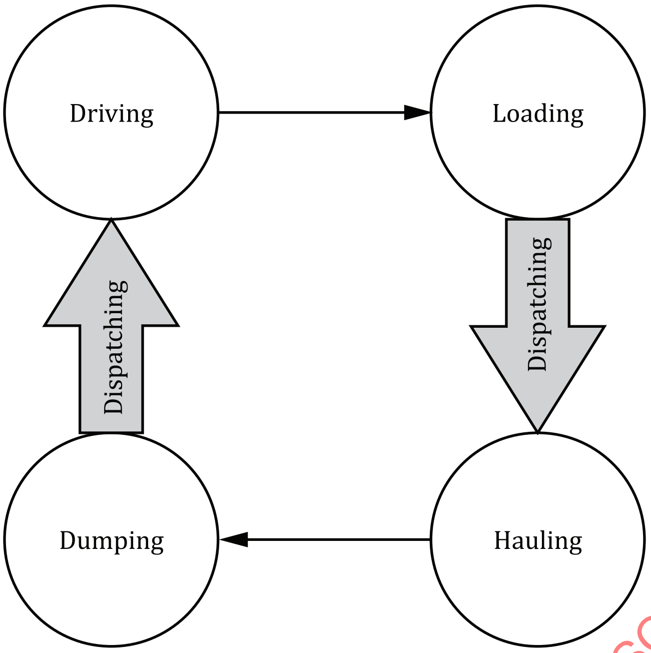


Figure 8 — Decision tree

Table 73 — Production state semantic

State	Meaning
Off	The machine‘s prime mover is not energized so the machine can’t do any work.
Driving	The machine is moving forward with its tray empty towards its assigned destination.
Spotting	The machine is moving slowly to a precise position to be ready to execute its task at the final destination (load, park, dump, fuel).
Loading	The machine is stopped and waiting for the loading machine to release the truck.
Hauling	The machine is moving forward with a payload in the tray towards its assigned destination.
Dumping	The machine does not have the tray down because it’s going through the tray emptying procedure.
Queued	The machine is stopped because there’s another truck in front preventing it from continuing with its assigned task.
Stopped	The machine is stopped because of a temporary event preventing the AHS from continuing with its assignment. Once the event is lifted, it intends on continuing the assignment.
Parked	The truck is meant to stay here for a while. The parking brake (service brake) is applied.

12.4 Haul truck production state message

AHS shall send haul truck production messages each time the production state of the truck changes.

Table 74 specifies which actor is allowed to send the message and when this message is expected.

Table 74 — Haul truck production state message

	AHS->FMS	FMS->AHS
At connection	No	No
Runtime change	Yes	No

The haul truck production message shall include the structure defined in Table 75. Generated when the production state changes. It will not when a new WayId is generated.

12.4.1 Haul truck production state V2

Table 75 — Haul truck production state V2

Member	Req. Level	Type	Description
HaulTruckProductionStateV2	shall	Object	Identifies this structure as Haul Truck production state object.
AssignmentId	shall	GUID	The Identification associated with the assignment the equipment is executing. If the machine does not posses an assignment, then this GUID **shall** be set to **null.**
WayId	shall	integer	WayId of the map object that the truck is currently in or on. If the truck is driving in an area that cannot be reconciled to a map object, then the WayId shall be set to **null.** If the WayId can resolve to multiple map objects, then it will follow the repeatable precedence order as defined by AHS. Only null or one WayID shall be sent.
AutonomyMode	shall	enumeration	Please see Table 8
ProductionState	shall	enumeration	Please see Table 74

```
{
 "Protocol":"ISO23725",
 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10.43.511Z",
 "EquipmentId": "2248d535-3daf-4a86-b1e1-4951a22beec6",

 "HaulTruckProductionStateV2": {
  "AssignmentId": "2248d535-3daf-4a86-b1e1-4951a22beec6",
  "WayId": 750321442,
  "AutonomyMode": "Manual",
  "ProductionState": "Hauling"
 },

 "MachinePositionV1": {
  "Heading": 211,
  "Latitude": 49.176854,
  "Longitude": -123.0718,
  "Elevation": 175.23 ,
  "LatitudeAccuracy": 0.31,
  "LongitudeAccuracy": 0.27,
  "HeightAccuracy": 0.58,
  "HeadingAccuracy": 2.1 ,
  "Speed": 44
 }
}
```

12.5 Haul truck production state sensor V1

AHS shall send haul truck production state sensor message stream.

Table 76 specifies which actor is allowed to send the message and when this message is expected.

Table 76 — Haul truck production state message

	AHS->FMS	FMS->AHS
At connection	No	No
Runtime change	Yes	No

The haul truck production message shall include the structure defined in Table 77

12.5.1 Haul truck production state sensor V1

The Truck Production State Sensor message shall be sent from the AHS:

- each time a digital sensor change state and
- each time an analog sensor changes significantly.
- With a configurable minimum period between each message

This message is a fallback solution if the truck production cycle can't be determined by the AHS.

Table 77 — Structure definition: Haul truck production state sensor V1

Member	Requirement Level	Type	Description
HaulTruckProduction StateSensorV1	shall	Object	Identifies this structure as Haul Truck production state sensor object.
AssignmentId	shall	GUID	The Identification associated with the assignment the equipment is executing. If the machine does not posses an assignment, then this GUID **shall** be set to **null.**
WayId	shall	Integer	WayId of the map object that the truck is currently in or on. If the truck is driving in an area that cannot be reconciled to a map object, then the WayId shall be set to **null.** If the WayId can resolve to multiple map objects, then it will follow the repeatable precedence order as defined by AHS. Only one WayID shall be sent.
AutonomyMode	shall	enumeration	Please see Table 8
GroundSpeed	shall	decimal	Speed at which the machine is moving in relation to the ground.
Payload	shall	percent	Rough percentage because this value is only used to determined if the truck is "Empty", "being loaded" or "Full". If the truck is equipped with a precise payload system, the truck payload message will also be sent.
EngineRunning	shall	boolean	Whether the machine’s power pack is ready to deliver energy to the machine.
Reversing	shall	boolean	If the prime mover and transmission configuration will make the machine go backwards.
TrayDown	shall	boolean	If the tray of the dump truck is in its resting position, ready to haul.
ParkingBrakeEngaged	shall	boolean	If the parking brake is engaged.

```
{
 "Protocol":"ISO23725",

 "Version": 1,
 "Timestamp": "2018-10-31T09:30:10.43.511Z",
 "EquipmentId": "2248d535-3daf-4a86-b1e1-4951a22beec6",

 "HaulTruckProductionStateSensorV1": {
  "AssignmentId": "0c83193f-8772-446a-89c0-a3977e282b8a",
  "WayId": 750321442,
  "AutonomyMode": "Manual",
  "GroundSpeed": 37.2 ,
  "Payload": 100,
  "EngineRunning": true,
  "Reversing": false,
  "TrayDown": true,
  " ParkingBrakeEngaged ": false
 },

 "MachinePositionV1": {
  "Heading": 211,
  "Latitude": 49.176854 ,
  "Longitude": -123.0718,
  "Elevation": 175.23 ,
  "LatitudeAccuracy": 0.31,
  "LongitudeAccuracy": 0.27,
  "HeightAccuracy": 0.58,
  "HeadingAccuracy": 2.1 ,
  "Speed": 0
 }
}
```

12.6 Haul truck payload message

Table 78 specifies which actor is allowed to send the message and when this message is expected.

Table 78 — Haul truck payload Messages initiator and trigger

	AHS->FMS	FMS->AHS
At connection	No	No
Runtime change	Yes	No

The haul truck payload message shall include the structure defined in Table 79

A haul truck payload message should be sent each time

— the payload of the truck increases significantly

— a final more accurate payload is determined

The payload message shall be timely initiated so the FMS can determine if the truck is overloaded before it reaches the haul road.