improving public transport through technology

Digital Air Interface Protocol

RTIG Library Reference: **RTIGT030-1.3**

March 2021

Availability: Members Only

Price:	
Foundation Members:	Free
Full Members:	Free
Associate Members:	Free
Non-members:	£250

© Copyright – RTIG Limited

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means, electronic, mechanical, photocopying or otherwise without the prior permission of RTIG Limited

No part of this document or of its contents shall be used by or disclosed to any other party without the express written consent of RTIG Limited

List of contents

Glossary of Terms

3G	Third Generation mobile services, generally backwards compatible with GPRS
ACK	“Acknowledged” (control code)
ATCO	ATCO Area Code
AVL	Automatic Vehicle Location
BCD	Binary Converted Decimal
CSV	Comma Separated Variable
ENQ	“Enquiry” (control code)
ETM	Electronic Ticket Machine
GPRS	General Packet Radio Service
GPS	Global Positioning System
Hex/HEX	Hexadecimal (base 16)
ID	Identity *or* identifier
IP	Internet Protocol
LA	Local Authority
LSB	Least Significant Bit
LVCC	Local Vehicle Control Centre
MSB	Most Significant Bit
NAK	“Not acknowledged” (control code)
NMEA	National Marine Electronics Association – in this context, the format in which GPS receivers present data on location etc
OBU	On Board Unit – namely, the device on the vehicle that communicates with the centre using this protocol. This may be an ETM or a separate unit
O-Licence Number	Operator Licence Number
PMR	Private Mobile Radio
RTI	Real Time Information
RTPI	Real Time Passenger Information
RTIG	Real Time Information Group

TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UTC	Coordinated Universal Time
VRN	Vehicle Registration Number
WiFi	Wireless Fidelity – local area radio LAN standards based on IEEE802.11
WGS84	World Geodetic System 1984 – standardised latitude-longitude reference system, as used by GPS

1 Introduction

1.1 This document

1.1.1 This document has been produced by RTIG as a specification for a set of messages that can be communicated between bus-mounted devices and office-based systems, using a digital open protocol.

1.1.2 The status of this document is **released**, as of March 2021**.** This issue is released as version v1.3. The document will be kept under configuration control.

1.1.3 Comments are made in the text to indicate how robust the Working Group feel that individual sections of this protocol specification are.

1.1.4 This document features extensive updates following the experience of a number of parties in implementing this specification. A number of the fields have been defined more accurately and a number of event messages have been added. Message structure and log on log off rules have also been updated.

1.2 Earlier versions

1.2.1 Version 1,2 clarified some aspects of the specification which were ambiguous or inconsistent in v1.1, specifically concerning:

- the coding of protocol version in the message wrapper and acknowledgement (sections 3.1 and 3.2)
- the fact that the Log-on Response is a reply Message and not an Acknowledgment (section 4.4 and worked example in annex B.2)
- differences in the coding of schedule deviation in Full and Short Position Updates (sections 4.9 and 4.10).

1.2.2 The first of these is significant in that, as proved by RTIG's Interoperability Test Day in March 2013, it could lead to non-working in the field.

1.2.3 New developments should no longer make use of v1.1. Suppliers offering products based on v1.1 should make it clear that this is an old version, and should aim to withdraw or update such products (including products in service if possible) by 31 December 2013.

1.2.4 This versions updates elements of traffic light priority in the Automatic Messages table to reflect updates in related standards and depreciates the extension for traffic light priority trigger locations with the standard RTIGT042 Traffic Light Priority Trigger Position File Format.

1.3 Context

1.3.1 Current real time systems use proprietary air interface protocols between bus and centre. Originally this practice was a result of the way that such systems were produced by the supply industry, but it was also necessary because of the need to ensure efficient use of limited radio spectrum. Spectrum efficiency remains a big concern with analogue Band III radio.

1.3.2 Circumstances are now changing, with a more mature and flexible systems industry and easier access to radiocomms bandwidth, including via digital channels. A key movement is one in which operators are establishing projects to provide GPRS connectivity to ticket machines. This provides a potential communications channel for conveying on-vehicle information back to centres, parallel to (or alternative to) separate RTI units.

1.3.3 However, ticket machines are provided by suppliers (almost always under contract to an operator) different from those providing RTPI system (almost always under contact to a local authority). A standard protocol will therefore be required for the interface.

1.3.4 This specification also supports the need for reliable connectivity with buses as vehicles move from one area to another, as well as taking the opportunity to implement some features which might be of use to operators only (i.e. providing data which is for operational control rather than public use).

1.4 Use Cases

1.4.1 The messages outlined in this specification can be used by different stakeholders in a variety of ways. Local authorities as well as bus operators will have the opportunity to identify, locate and check on the status of their vehicles throughout their period of service.

1.4.2 This specification could also assist bus operators improving the accuracy of bus service information. Operators themselves can benefit through the provision of detailed vehicle and driver information, while also providing service users with more accurate RTI data while a bus is in service, as well as improving the accuracy of bus timetabling.

1.4.3 This specification may also be used to improve bus service efficiency; contributing to the effectiveness of traffic light priority systems, as well as providing centres with road status updates (e.g. accident and congestion reports).

1.4.4 This specification is therefore relevant to:

- Bus operators looking to develop their RTI messaging systems so that they can regularly locate their vehicles throughout the UK.

- Ticket machine suppliers looking to implement such a standard message set into their machines.
- Other suppliers for buses looking to standardise their messaging systems across bus operators.
- Local authorities looking to standardise their bus location messages so that they can be used across local authority boundaries.

1.4.5 Note that this specification does not impose functional requirements on the system. Issues such as required location accuracy, communications availability etc must be specified separately.

1.5 Scope

1.5.1 This specification does not attempt to duplicate all possible functions of an RTI OBU. It concentrates, rather, on the definition of a basic, common set of messages that would meet most common needs.

1.5.2 The core of the specification is presented in four message sets (section 4):

- **Log On/Off**
  - **log on** (Log On Request and Log On Response messages): provides a notification that a vehicle is powering on and about to begin a period of service. This message contains details of the identity of the vehicle. Following the receipt of this Log On message by the appropriate centre, a unique identifier is issued to the vehicle to be used for that particular period of service.
  - **log off** (Log Off Request): provides a notification that a vehicle has ended a period of service and is powering down. This message contains the session identifier for the vehicle.
- **Journey**
  - **journey initiation** (Journey Details message): outlines the parameters of the journey currently being delivered by a vehicle in service. This message should be sent once for each journey and will always be accompanied by a Positional Update Message.
  - **end of journey** (End of Journey message): provides a notification that a vehicle is about to end a particular journey. This differs from the Log Off message as this provides a notification for the end of a period of service.
- **Position**
  - **position update** (Position Update message): provides a regular positional update message while a vehicle is in operation.
- **Events**

- a range of **events** based on a common Events message frame, parameterised for context: this includes emergencies, vehicle status messages, driver information messages, etc.

1.5.3 Of these, the first three are considered fairly robust and may be used with a reasonable degree of confidence. The Events messages are still under development and are considered to be more likely to undergo change. Additional message types may also be added in future.

1.5.4 The specification also includes a number of common message structuring, framing and coding rules (sections 2 and 3).

1.5.5 It is anticipated that guidance will be developed on operational aspects of using this protocol which are not suitable for standardisation at this point, e.g. how to manage update frequency and timeout thresholds.

1.5.6 These messages are designed for use in systems using IP-compatible communications, such as GPRS, 3G and WiFi mesh.

2 Basis of specification

2.1 Architecture

2.1.1 It is assumed that the device on bus sending data has intelligence and a knowledge of schedule information derived from a TransXChange source. Hence it can send information on the journey and operator. Therefore the production of a standard schedule configuration file for the vehicle is out of scope of this standard.

2.1.2 In the absence of this on bus intelligence only a basic feature set will be supported based on returning journey details as entered by the driver. Note also that the sending device will also have to be configured for operator, vehicle ID etc.

2.1.3 UDP/IP is used rather than TCP/IP for communications protocol as it has a lower overhead (28 bytes per message – 20 bytes IP header, 8 bytes UDP header). This may impose operational requirements for trapping non-communicating units.

2.1.4 Vehicle IP addresses should be identified from the source IP address information in the IP header when a vehicle logs on. Once the vehicle session ID has been issued, this may be used to trap changes in IP address (e.g. if a vehicle falls out of GPRS coverage and is subsequently given a new dynamic IP address). Alternatively, the vehicle may be required to log in again, and be given a new session ID.

2.1.5 The same port should be used for sending and receiving UDP telegrams. The default port and IP address to be used needs to be configured into the OBU.

2.1.6 It is required that the optional 16-bit checksum field will be included in the UDP Header. The receiver will be required to verify the checksum. Note that UDP uses ones-complement arithmetic, so a computed zero value is stored in the UDP Header as all-ones.

2.2 Mandatory and optional fields

2.2.1 Within each message, a number of parameters have been outlined which will form the structure of the message itself. These parameters are described as either:

- Mandatory: parameters which must be included in messages for every RTPI scheme.
- Optional: parameters which may be included subject to the specification of the stakeholders involved.

2.2.2 For example, an operator may specify some of the “optional” parameters for its ticket machine contractor (which must then be delivered under contract), but may not assume that every LA area in which the vehicle runs will be able to handle those parameters.

2.2.3 In the case of the Log on, Journey Details and Positional Update messages, all parameters have been allocated a mandatory status in order to fix the length of these messages. This was carried out in order for a clear price structure for users. In situations where users do not use particular parameters within a message, many of the parameters can be populated with null values.

2.3 Acknowledgements

2.3.1 An acknowledgement for a message is requested by setting the Acknowledgment flag in the wrapper. Section 4 indicates the acknowledgment requirements for each message type – when sending a message of a given type, the Acknowledgment flag must be set accordingly.

2.3.2 Where more than one message is contained within a single wrapper, the Acknowledgement flag applies only to the first message – messages after the first can only be of a type which does not require an acknowledgement.

2.4 Efficiency of Communication

2.4.1 The efficiency of communicating these messages has been a major point of discussion when producing this specification. Due to their use of a narrower bandwidth, binary numbers are used where possible instead of ASCII characters. The result is a reduction in message size and therefore communications cost. A similar argument was also made for the use of numeric characters as opposed to alphanumeric ones in a bid to reduce the price (numeric codes can be squeezed two characters to a byte, where alpha codes need a byte per character).

2.4.2 A unique Session Vehicle Identifier is used to improve the efficiency of the sending the messages contained within this specification. Instead of a vehicle being required to send its vehicle and operator ID for each message sent, this information will only be contained within the Log On message sent at the beginning of a period of service. Following the receipt of this Log On message by the appropriate centre, the unique identifier will be issued to the vehicle, to be used for any subsequent messages sent.

2.4.3 Aside from this coding efficiency, no provision has been made for the use of data compression algorithms. The rationale is that:

- The benefit of this last few percent of bandwidth efficiency is likely to modest.
- It would impose an avoidable processing overhead on systems.

2.4.4 While the specification allows for multiple messages to be combined in one wrapper, this may not always be possible because of acknowledgement requests. However, it is possible to combine multiple wrappers in the same UDP frame.

2.5 Field Coding

2.5.1 With the exception of Event Messages, all fields are fixed length. Fields may be one of the following Field Types:

- C = Alphanumeric character (ASCII, encoded one to a byte), with no leading spaces and with a terminating null (Hex 00) provided there is sufficient space[1]
- N = Nibble occupying 4 bits representing either a BCD numeric digit in the range 0-9 or as up to four individual binary flags; '0' or '1' (encoded two to a byte)
- X = Hexadecimal value representing a binary numeric value in the range 0-255, or up to eight individual binary flags '0' or '1' (encoded as one byte) [2]
- S = Binary Signed – 2's Complement Integer
- U = Binary Unsigned Integer

2.5.2 Where allowable and appropriate, null values will be as follows:

- For C fields, a sequence of Hex 00;
- For numerical N fields, a sequence of Hex F (a non-existent numeric)
- For numerical X fields, a sequence of Hex FF (equivalent to the value -1)
- For Signed Integer there is no Null Value an invalid value may however be defined for specific fields.
- For Unsigned Integer the maximum value is treated as NULL (e.g. 0xFFFF)

Bitfield (flag) fields, either N or X, are never permitted to be null.

---

1 In fixed length C fields, the convention is that characters after the null termination are ignored and so any value may be inserted. In practice a sequence of final nulls is likely to be safest, to avoid misinterpretation.

2 Bits not currently used should normally be defaulted to 0.

2.5.3 The following special codes are used, as sequences (normally pairs) of N type (nibble) digits:

- h = Hour (24 hour clock)
- m = Minute
- s = Second
- D = Day
- M = Month
- Y = Year

2.5.4 Parameters are sent in normal reading order, ie first character or most significant byte first ("Big Endian"). For example:

- Public Service Code 15A, with format CCCCCC, would be transmitted as bytes 31, 35, 41, 00, 00, 00 (Hex).
- Format version 1.23, with format NNNN, would be transmitted as two bytes 01, 23 (Binary coded decimal).
- Direction – Inbound ('1'), with format X, would be transmitted as one byte, 01 Hex (or equivalently 00000001 binary).
- Format XXXX (0xFECD3523) would be transmitted as 4 bytes 0xFE, 0xCD, 0x35, 0x23.

2.5.5 Field values marked "not used" etc should not be used for vendor-specific extensions, to avoid difficulties arising with systems built to future versions of this specification.

3 Message basics

3.1 Wrapper Structure

3.1.1 Messages using this specification shall have the following wrapper structure.

Parameter	Mandatory/ Optional	Null Value Allowed?	Format (Byte Aligned)
Format Version No.	Mandatory	No	NNNN *Stored as Nibbles and refers to the version of this specification used, e.g. 0122 means Version 01.22 and 0070 means Version 00.70* *Version numbers are regarded as right aligned, e.g v3.7 should be interpreted as Version 03.07 and coded as 0307*
Message Flags	Mandatory	No	X (bit mapped)
Message Counter	Mandatory	No	XX
Session Vehicle Identifier	Mandatory	No	XX
Optional Data Fields	Mandatory	No	XX (bit mapped) *e.g. 1111 0110 0xxx xxxx, where x = Don't care*
Message Payload	Mandatory	No	Mandatory and Optional data fields
Message Time Stamp	Mandatory	No	XXXXXX Encoded as BCD: YYMMDDhhmmss *i.e. six bytes*

3.1.2 The Message Structure is applicable to Outgoing and Incoming messages (with respect to the bus-based device) and is structured in such a way as to allow for message enhancements and optional data fields.

3.1.3 *Format Version Number*: This parameter uniquely identifies the version number of this protocol (ie of this document) that is being used. The Format Version Number shall be amended to reflect alterations or enhancements to the payload. The Format Version Number requires four Nibbles in BCD format representing two-digit major and two-digit minor version number in the form 00.01 to 99.99 with 00.00 being reserved.

3.1.4 *Message Flags*: The Message Flags shall be used to qualify the Message in regard of Acknowledgement Requests and Acknowledgements. The ability to identify Test Messages within a live system is supported. This parameter requires a single Nibble and is bit mapped as detailed below.

- Bit 0 – always 0, “This is a Message” (see **Acknowledgement Structure**)
- Bit 1 – Require Acknowledge Flag (If Bit 0 = 0)
  - 0 = No Acknowledgement required
  - 1 = Acknowledgement required
- Bit 1 – Message Acknowledgement Flag (If Bit 0 = 1)
  - 0 = Message is not Acknowledged (Error in message)
  - 1 = Message is Acknowledged
- Bit 2 – Reserved
- Bit 3 – Live or Test/Evaluation Message
  - 0 = Live Message
  - 1 = Test Message
- Bits 4-7 – Reserved.
  - Identifies Multiple Message transmission status:
    - 0000 - Multiple Message transmission **NOT** implemented
    - 0001 - 1111 – Message Wrapper contains between 1 (one) and 15 (fifteen) concatenated messages.

3.1.5 *Message Counter*: The Message Counter will be under the control of the Message Sender and will provide a unique Message Reference that may be used in subsequent messages if reference to previous messages is required. The Message Counter shall be in the range 0-65535 and will wrap to 0 (Zero) following terminal count increment. New sessions should be default initiate a new Message Counter sequence from 0.

3.1.6 *Session Vehicle Identifier*: The Session Vehicle Identifier parameter is derived from the Log On Response as a result of the Log On Request generated by the OBU. For the initial Log On Request message, when this is clearly unavailable, this parameter shall be set to 0 (zero). This parameter requires two bytes in the range 1-65535 where 0 (zero) is reserved as described above.

3.1.7 *Optional Data Fields*: The Optional Data Fields parameter describes which if any of the Optional Parameters are contained within the Payload of the message. Where message types and corresponding message payload do not support Optional parameters, then this parameter shall be ignored. The Optional Data Fields parameter requires two bytes in 16 bit format where each bit corresponds to an optional data parameter. The 16 bits are ordered MSB to LSB where the most significant bit corresponds to the first optional parameter. Unused bits ("don't cares") shall be set to 0 (zero). Where an optional field is used, the message must be sent in its own message wrapper.

3.1.8 *Message Payload*: The Message Payload contains the Mandatory and Optional data fields for the given message type. The ability to accommodate Multiple Messages within the Message Payload is supported as defined by Message Flags bits 4 – 7. The message types are described below.

3.1.9 *Message Time Stamp*: The Message Time Stamp parameter shall be the GPS's NMEA 0183 representation of its Coordinate Universal Time (UTC). This parameter will prove useful when network delays are encountered.

3.2 Acknowledgement Structure

3.2.1 Acknowledgements using this specification shall have the following structure.

Parameter	Mandatory/ Optional	Null Value Allowed?	Format (Byte Aligned)
Format Version No.	Mandatory	No	NNNN *Stored as Nibbles and refers to the version of this specification used, e.g. 0122 means Version 01.22 and 0070 means Version 00.70* *Version numbers are regarded as right aligned, e.g. v3.7 should be interpreted as Version 03.07 and coded as 0307*
Acknowledgement Flags	Mandatory	No	X (bit mapped)
Acknowledgement Counter	Mandatory	No	XX
Message Counter Reference	Mandatory	No	XX *Previous Incoming Message Ref.*

Parameter	Mandatory/ Optional	Null Value Allowed?	Format (Byte Aligned)
Session Vehicle Identifier	Mandatory	No	XX
Message Time Stamp	Mandatory	No	XXXXXX Encoded as BCD: YYMMDDhhmmss *i.e. six bytes*
Error Number	Mandatory	No	X As defined below

3.2.2 The Acknowledgement Structure may be required in response to the previous Incoming Message generated by either the OBU or the Central Communication Server and is structured as follows.

3.2.3 *Format Version Number*: This parameter uniquely identifies the version number of this protocol (i.e. of this document) that is being used. The Format Version Number shall be amended to reflect alterations or enhancements to the payload. The Format Version Number requires four Nibbles in BCD format representing major and two digit minor version number in the form 00.01 to 99.99 with 00.00 being reserved.

3.2.4 *Acknowledgement Flags*: The Acknowledgement Flags shall be used to qualify the Acknowledgement in relation to the previous Incoming Message. The ability to identify Test Messages within a live system is supported. This parameter requires a single Nibble and is bit mapped as detailed below.

- Bit 0 – always 1, “This is an Acknowledgement” (see **Message Structure**)
- Bit 1 – Message Acknowledgement Flag
  - 0 = Message is not Acknowledged (Error in message)
  - 1 = Message is Acknowledged
- Bit 2 – Reserved
- Bit 3 – Live or Test/Evaluation Message
  - 0 = Live Message
  - 1 = Test Message
- Bits 4 – 7 – Reserved.

3.2.5 *Acknowledgement Counter*: The Acknowledgement Counter will be under the control of the Acknowledgement Sender and will provide a unique message reference. The Acknowledgement Counter shall be in the range; 0-65535 and will wrap to 0 (Zero) following terminal count increment.

3.2.6 *Message Counter Reference*: The Message Counter Reference is used by the Acknowledgement to reference the previous Incoming Message from either the OBU or the Central Communication Server. This is a Mandatory parameter and will be in the range 0-65535.

3.2.7 *Session Vehicle Identifier*: The Session Vehicle Identifier parameter is derived from the Log On Response as a result of the Log On Request generated by the OBU. For the initial Log On Request message, when this is clearly unavailable, this parameter shall be set to 0 (zero). This parameter requires two bytes in the range 1-65535 where 0 (zero) is reserved as described above.

3.2.8 *Message Time Stamp*: The Message Time Stamp parameter may be derived from the GPS's NMEA 0183 representation of its Coordinate Universal Time (UTC) (if sent by the OBU) or the server time (if sent by the server). This parameter will prove useful when network delays are encountered.

3.2.9 *Error Number*: This field is used to send an error code to indicate the reason for a negative acknowledgement. **The field is ignored for a positive acknowledgement.** The following are some standard codes, however it is expected that application specific codes may be required.

Number	Meaning
0	No error code
1	Unknown Sender – e.g. SVID is unknown
2	Unknown Running Board
3	Unknown Operator Code
4	Unknown Vehicle ID
5	Unknown Service Code
6	Unknown Journey Number
7	Unknown Event Received
8	Unknown Event data sent with known Event
9	Config Information not supported
10	Serial No Information not supported
11	Duplicate Vehicle ID
12	No SVID available
13	Corrupt Message Received
14 .. 127	Reserved codes
128..255	Implementation defined codes

4 Message structures

4.1 Overview

4.1.1 As displayed in the table below, a number of messages have been defined, with the vast majority allocated a 'frozen' status. It was deemed by document contributors that these frozen messages were in a suitable state to be used. The Events messages however are yet to be finalised and still in a 'draft' format. Further work is therefore required before draft messages are produced in a final format.

4.1.2 In addition, each message has been assigned a category: either Principal or Variant. The Principal messages are those containing all of the relevant parameters for a particular message. The Variant messages provide different parameter combinations, enabling users to choose which set of parameters is most appropriate for them. For example, in the case of both the Journey Details and Position Update messages 'basic' messages have been used as Variant messages where only the minimum number of essential parameters are included.

4.1.3 The following message types are currently defined.

Message ID	Message Type	Category	From	Message Status
10	Log On Request	Principal	OBU	Frozen
11	Log Off Request	Principal	OBU	Frozen
20	Log On Response	Principal	Server	Frozen
30	Journey Details	Principal	OBU	Frozen
31	Journey Details (basic)	Variant	OBU	Frozen
32	Journey Details (enhanced)	Variant	OBU	Not yet specified
39	End of Journey	Principal	OBU	Frozen
40	Position Update	Principal	OBU	Frozen
41	Position Update (basic)	Variant	OBU	Frozen
42	Position Update (enhanced)	Variant	OBU	Not yet specified
50	Event (OBU to centre)	Principal	OBU	Draft
60	Event (centre to OBU)	Principal	Server	Draft
255	Enquiry	Principal	Server	Draft

4.2 Message Acknowledgement

4.2.1 The structure of the Acknowledgement Message is defined in section 3.

4.2.2 A Message Acknowledgement may be sent by either the Server or the OBU in response to a previously received Message where the associated Message Flags specifically request a response.

4.2.3 A Message Acknowledgement does not itself require an acknowledgement and thus the Acknowledgement Flags do not support this requirement.

4.3 Log On Request – Message ID #10

4.3.1 NB the Log On Request message relates to a vehicle logging on to the RTI system and not to the driver logging on to the ETM.

This Message Type does not require an acknowledgement

Parameter	Mandatory/ Optional	Null Value Allowed?	Format	Length (bytes)
Message ID	Mandatory	No	10 (0A Hex)	1
Operator ID	Mandatory	No	CCCCCCCCC *O-licence number*	9
Vehicle ID	Mandatory	No	CCCCCCC *From existing LVCC system*	7
OBU ID Length	Optional	No	X. Length of the OBU ID field	1
OBU ID	Optional	No	CCC... Some unique identifier for example model number and serial number combined.	n

4.3.2 *Message ID*: This parameter uniquely defines the message type. The Message ID requires a single byte capable of describing 255 (1-255) message types with 0 (Zero) being reserved.

4.3.3 *Operator ID*: The Operator ID parameter is derived from the Operator O-Licence number and forms part of a vehicle's unique identifier along with the Vehicle ID parameter.

4.3.4 *Vehicle ID*: The Vehicle ID parameter is derived from the vehicle's fleet number and forms part of a vehicle's unique identifier along with the Operator ID parameter. Unlike the Operator ID, the fleet number is only unique within an operator and may conflict with other UK operators. Smaller operators without a separate fleet numbering system may choose to use the VRN for this field.

4.3.5 *OBU ID Length*. This defines the length of the following field.

4.3.6 *OBU ID*: This is a unique identifier for the OBU. The format is manufacturer specific.

4.3.7 This message is sent from the vehicle to the control centre. The Control centre should respond with Log On Response message.

4.3.8 This message is always sent with the SVID in the wrapper equal to 0.

4.4 Log On Response – Message ID #20

This Message Type does not require an acknowledgement

Parameter	Mandatory/ Optional	Null Value Allowed?	Format	Length (bytes)
Message ID	Mandatory	No	20 (14 Hex)	1
Session Vehicle Identifier	Mandatory	No	XX A returned value of 0 (zero) indicates an invalid Log on.	2
Error Number	Mandatory	No	X See Acknowledgement Error code	1
Server Address	Optional	No	XXXX IP address of the server to which data is to be addressed rather than the default address	4
Server Port	Optional	No	UU IP Port on the server to send information to.	2

4.4.1 *Message ID*: This parameter uniquely defines the message type. The Message ID requires a single byte capable of describing 255 (1-255) message types with 0 (Zero) being reserved.

4.4.2 *Session Vehicle Identifier*: The Session Vehicle Identifier parameter is in response to the Log On Request message generated by the OBU and managed by the Scheme centre. This parameter will be used in all subsequent messages as a means of identifying specific vehicles. This parameter requires two bytes and the returned value will be in the range 1-65535. It must also be noted that multiple IP addresses can be used within a single session. The SVID in the wrapper should be ignored until a valid SVID has been received in the Log On Response.

4.4.3 *Error Number*: Although this message is not an “acknowledgement”, it is convenient to include an error field that uses the same conventions as acknowledgement errors (see Section 3.2.9). For instance, if the server receives a Log On Request and it does not recognise the Operator ID, the SVID will be 0 (“Invalid Log on”) and the Error Number field will be 3 (“Unknown Operator Code”).

4.4.4 *Server Address*: the address to which the server would prefer the mobile to send data. The intention of this is to allow the server to perform load balancing across the multiple servers. If the Server Address is omitted or is not supported by the OBU then the original configured base address to which the log on requires was sent should continue to be used by the OBU.

4.4.5 *Server Port*: the port on [Server Address] to which data is to be sent from the OBU.

4.4.6 The SVID is allocated by the server on the first logon on request received. This should be reused for the same vehicle (as identified by vehicle id , operator id) on each subsequent logon request received before a logoff message from the OBU is received or the server “times-out” the session id. The message counter should not be reset if the same vehicle requests a log on when an SVID is already allocated to it. The optional Serial Number can be used to validate the uniqueness of the OBU requesting a connection and allows the server to reject duplicated vehicle IDs.

4.4.7 If this message is received by the OBU and the SVID in the telegram is different to the SVID the OBU currently has then the OBU should reset the message counter and use the new SVID in all subsequent communications.

4.5 Log Off Request – Message ID #11

This Message Type does require an acknowledgement

Parameter	Mandatory/ Optional	Null Value Allowed?	Format	Length (bytes)
Message ID	Mandatory	No	11 (0B Hex)	1
Session Vehicle Identifier	Mandatory	No	XX	2

4.5.1 *Message ID*: This parameter uniquely defines the message type. The Message ID requires a single byte capable of describing 255 (1-255) message types with 0 (Zero) being reserved.

4.5.2 *Session Vehicle Identifier*: The Session Vehicle Identifier parameter is derived from the Log On Response as a result of the Log On Request generated by the OBU. This parameter requires two bytes in the range 1-65535 where 0 (zero) is reserved as described above.

4.5.3 When this message is received at the control centre, this vehicle is to be recorded as logged off and no further communication is to be received against this SVID until some other login results in the SVID being reallocated. Receipt of a message using an unallocated SVID and requiring acknowledgement should result in the Server sending a NACK to the vehicle with the *Error Number* 1. The OBU should then follow the log on process.

4.6 Journey Details – Message ID #30

Null parameters for Message ID #30 and #31

4.6.1 Care is needed with the Journey Details message, where varying operational practice makes it necessary to identify the journey uniquely in a number of different ways. The following are some of these (non-exclusive):

- Journey Scheduled Start Time is not null, but Journey Number is null
- Journey Number is not null, but Journey Scheduled Start Time is null

4.6.2 In engineering a system, the bus system provider and centre system provider must agree on the business rules to be adopted, in order to allow message integrity checking and parsing.

This Message Type does require an acknowledgement

Parameter	Mandatory/ Optional	Null Value Allowed?	Format (Byte Aligned)	Length (bytes)
Message ID	Mandatory	No	30 (1E Hex)	1
Service Code	Mandatory	No	CCCCCC	6
Running Board	Mandatory	Yes	CCCCCCC *Operator defined*	7
Journey Number	Mandatory	Yes	CCCCC *Operator defined*	5
Journey Scheduled Start Time	Mandatory	Yes	XX Encoded as BCD: hhmm *Referenced to current time zone (including summer time)*	2
Duty Number	Mandatory	Yes	CCCCCC *Operator defined*	6
Public Service Code	Mandatory	No	CCCCCC	6
Direction	Mandatory	No	X	1
Depot Code	Mandatory	Yes	CCCC *Operator defined*	4
Driver ID	Mandatory	Yes	CCCCCC	6
First stop ID	Mandatory	Yes	CCCCCCCCCCCC *AtcoCode (from NaPTAN)*	12
Destination Stop ID	Mandatory	Yes	CCCCCCCCCCCC *AtcoCode (from NaPTAN)*	12

4.6.3 *Message ID*: This parameter uniquely defines the message type. The Message ID requires a single byte capable of describing 255 (1-255) message types with 0 (Zero) being reserved.

4.6.4 *Service Code*: The Service Code parameter will normally be derived from the Electronic Ticket Machine and will correspond to the TransXChange TicketMachineServiceCode.

4.6.5 *Running Board*: The Running Board parameter will normally be derived from the ETM, and will correspond to the TransXChange BlockNumber. It may be used to help distinguish multiple journeys with the same start time.

4.6.6 *Journey Number*: The Journey Number parameter will normally be derived from the ETM, and will correspond to the TransXChange VehicleJourneyCode. It may assist in uniquely identifying the subject vehicle's journey.

4.6.7 *Journey Scheduled Start Time*: The Journey Scheduled Start Time parameter will be derived from the ETM, and will correspond to the TransXChange Scheduled Departure Time. The scheduled start time relates to the first in-service stop of the journey, even if drivers have changed in the middle of a journey.

4.6.8 *Duty Number*: The Duty Number is derived from the ETM and will correspond to the TransXChange DutyCrewCode.

4.6.9 *Public Service Code*: The Public Service Code parameter may be available from the ETM (and corresponds to TransXChange Line Name). Often it will be the same as the Service Code, but there may be minor variations for operational reasons, and the Public Service Code may not always be available.

4.6.10 *Direction*: The Direction parameter is derived from the ETM if it is adopted by operators. Possible confusion could arise however if the route is being shared by more than one service. Typically the following convention should be used: '0' = OUT, '1' = IN, '2' = Circular Clockwise and '3' = Circular Anti-Clockwise.

4.6.11 *Depot Code*: The Depot Code parameter is derived from the ETM and will correspond to the TransXChange GarageCode. If required, it may be used to identify a service if several depots operate when there are services with the same public service code.

4.6.12 *Driver ID*: The Driver ID is derived from the ETM.

4.6.13 *First Stop ID*: The First Stop ID is derived from the AtcoCode field of NaPTAN and, if adopted, could be used as a route identifier when non-unique journey start times occur. Similarly the *Destination Stop ID* is derived from the AtcoCode.

4.6.14 Although not specified in the message structure, the Journey Details Message will always be accompanied by a Position Update Message (see Message ID #40).

4.7 Journey Details (basic) – Message ID #31

This Message Type does require an acknowledgement

4.7.1 This message is a cut-down version of the Journey Details message (ID #30) which includes only a basic set of data. As well as being shorter, all of its fields are mandatory, which simplifies database processing.

4.7.2 Where data is not available a field of valid nulls shall be sent, eg in the absence of a Journey Scheduled Start Time the bytes FF FF (Hex) shall be sent.

Parameter	Mandatory/ Optional	Null Value Allowed?	Format	Length (bytes)
Message ID	Mandatory	No	31 (1F Hex)	1
Service Code	Mandatory	No	CCCCCC	6
Running Board	Mandatory	Yes	CCCCCCC *Operator defined*	7
Journey Number	Mandatory	Yes	CCCCC *Operator defined*	5
Journey Scheduled Start Time	Mandatory	Yes	XX Encoded as BCD: hhmm *Referenced to current time zone (including summer time)*	2
Duty Number	Mandatory	Yes	CCCCCC *Operator defined*	6
Public Service Code	Mandatory	No	CCCCCC	6
Direction	Mandatory	No	X	1

4.8 End of Journey – Message ID #39

This Message Type does require an acknowledgement

4.8.1 The End of Journey message type is sent by the OBU having completed the entire Journey identified in Message Types #30 or #31. For premature termination of the current Journey, then Event Message Type #50 Message Code #4, Abandoning Journey, may be sent with optional free text description.

4.8.2 Where data is not available a field of valid nulls shall be sent, e.g. in the absence of a Journey Scheduled Start Time the bytes FF FF (Hex) shall be sent.

4.8.3 NB Where it is operationally difficult to configure the system to send this message, the server may regard it as having been sent on receipt of a valid new Journey Details message from the same vehicle where the journey details are different to the current journey known to the Server.

Parameter	Mandatory/ Optional	Null Value Allowed?	Format	Length (bytes)
Message ID	Mandatory	No	39 (27 Hex)	1
Service Code	Mandatory	No	CCCCCC	6
Journey Number	Mandatory	Yes	CCCCC *Operator defined*	5
Journey Scheduled Start Time	Mandatory	Yes	XX Encoded as BCD: hhmm *Referenced to current time zone (including summer time)*	2
Public Service Code	Mandatory	No	CCCCCC	6
Direction	Mandatory	No	X	1

4.9 Position Update Message – Message ID #40

This Message Type does not require an acknowledgement

Parameter	Mandatory/ Optional	Null Value Allowed?	Format	Length (bytes)
Message ID	Mandatory	No	40 (28 Hex)	1
Position – latitude	Mandatory	No	SSSS *Milliarcsecs north (negative for south of equator)* *0x7FFFFFFF indicates invalid or unknown*	4
Position – longitude	Mandatory	No	SSSS *Milliarcsecs east (negative for west of Greenwich Meridian)* *0x7FFFFFFF indicates invalid or unknown*	4
Bearing	Mandatory	No	U *Compass bearing in degrees divided by 2, e.g. East is 45* *0xFF indicates bearing unknown.*	1
Number of satellites visible	Mandatory	No	X 0 = No GPS Fix	1

Parameter	Mandatory/ Optional	Null Value Allowed?	Format	Length (bytes)
Position Quality	Mandatory	No	NN *Bit 0..3 – Quality of last stop* *Bit 4..7 = Quality of GPS* ***Values:*** *0 = Not Valid,* *1 = Valid* *Value > 1 scheme specific to be treated as valid.*	1
Last stop Index	Mandatory	Yes	U *The count of stops along the route. 0 = not reach first. N the number of the last stop counting from 1 as defined by the stop pattern (not just bus stopping)*	1
Distance travelled from last stop	Mandatory	Yes	XX *In metres*	2
Schedule Deviation	Optional	Yes	S Schedule Deviation in multiples of 30 seconds 0 = 0 0x01 to 0x7F Max pos 0xFF to 0x81 Max neg 0x80 is unknown	1

4.9.1 *Message ID*: This parameter uniquely defines the message type. The Message ID requires a single byte capable of describing 255 (1-255) message types with 0 (Zero) being reserved.

4.9.2 *Position*: The Position parameter is derived from WGS84 latitude and longitude values and, dependent on operator preferences, is calculated to varying degrees of accuracy.

4.9.3 *Bearing*: The bearing parameter is derived from the GPS feed if it is adopted by operators, using an NMEA format.

4.9.4 *Number of Satellites*: The number of satellites parameter is derived from vehicle GPS information to identify the number of satellites providing GPS location data.

4.9.5 *Message quality*: Optional parameter, set and used by agreement within a scheme. For instance it might incorporate information on number of GPS satellites, availability and currency of updates from door closures, gyros and dead reckoning, or equipment health indicators.

4.9.6 *Last stop ID*: Optional parameter, set and used by agreement within a scheme. As a single byte, it is intended to refer to the stop number in the stop sequence that constitutes the route, rather than a full NaPTAN ID.

4.9.7 *Distance travelled from last stop*: The distance travelled from last stop as derived from any available on-vehicle systems (GPS, odometer dead reckoning, beacons etc). A value of 0 here may be used to indicate that the vehicle has reached its final stop.

4.9.8 *Schedule Deviation*. This is an optional value to be sent if the OBU has a timetable and is calculating the schedule deviation along the journey.

4.10 Position Update Message (basic) – Message ID #41

This Message Type does not require an acknowledgement

4.10.1 Similarly to Message ID #31, this message is a cut-down version of the Position Update message (ID #40) which includes only a basic set of data.

4.10.2 Where data is not available a defined value is to be sent, e.g. in the absence of a Bearing the byte FF (Hex) shall be sent.

Parameter	Mandatory/ Optional	Null Value Allowed?	Format	Length (bytes)
Message ID	Mandatory	No	41 (29 Hex)	1
Position – latitude	Mandatory	No	SSSS *Milliarcsecs north (negative for south of equator)* *0x7FFFFFFF indicates invalid or unknown*	4
Position – longitude	Mandatory	No	SSSS *Milliarcsecs east (negative for west of Greenwich Meridian)* *0x7FFFFFFF indicates invalid or unknown*	4
Bearing	Mandatory	No	U *Compass bearing in degrees divided by 2, e.g. East is 45* *0xFF indicates bearing unknown.*	1
Schedule Deviation	Optional	No	S Schedule Deviation in multiples of 30 seconds 0 = 0 0x01 to 0x7F Max pos 0xFF to 0x81 Max neg 0x80 is unknown	1

4.11 Event (OBU to centre) – Message ID #50

This Message Type does require an acknowledgement

Please note that this Message Type is likely to be the subject of considerable further development.

Parameter	Mandatory/ Optional	Null Value Allowed?	Format	Length (bytes)
Message ID	Mandatory	No	X	1
Sequence ID	Mandatory	No	XX	2
Reference Sequence ID	Mandatory	No	XX	2
Position – latitude	Mandatory	No	SSSS *Milliarcsecs north (negative for south of equator)* *0x7FFFFFFF indicates invalid or unknown*	4
Position – longitude	Mandatory	No	SSSS *Milliarcsecs east (negative for west of Greenwich Meridian)* *0x7FFFFFFF indicates invalid or unknown*	4
Message Type	Mandatory	No	X *See table below*	1
Message Code	Mandatory	No	X *See table below*	1
Message Parameters Length	Optional	No	X *See table below*	1
Message Parameters Data	Optional	No	Qualifying Data	1-255

4.11.1 *Message ID*: This parameter uniquely defines the message type. The Message ID requires a single byte capable of describing 255 (1-255) message types with 0 (Zero) being reserved.

4.11.2 *Sequence ID*: The Sequence ID will be generated by the OBU and will provide a unique reference for each event message sent. This reference will be used in subsequent messages if communications centres wish to make further enquiries regarding a specific event. This is a Mandatory parameter and shall be in the range; 1-65535 with 0 (Zero) being reserved.

4.11.3 *Reference Sequence ID*: The Reference Sequence ID may be used by the OBU in order to reference a previous Incoming Message from the communications centre. This is a Mandatory parameter and will be in the range 1-65535. If the reference to a previous Incoming Message is not required then this field will be set to 0 (Zero).

4.11.4 *Position Identifiers* (latitude and longitude): The Position parameters are derived from WGS84 latitude and longitude values and, dependent on operator preferences, are calculated to varying degrees of accuracy. Alternative positioning systems, that could be used, include the existing national grid coordinate or the Ordnance Survey coordinate systems.

4.11.5 *Message Type Identifier*: The Message Type Identifier parameter will be derived from the OBU and describes the Group type of the outgoing message. At present, a limited range of Group types have been assigned and additional Groups may be included at a later date. This is a Mandatory parameter and will be in the range 0-255. The Message Type parameter supports two sub-ranges:

- 0-127 (Manual Messages);
- 128-255 (Automatic Messages).

Message Types 0 and 128 have the highest priority.

4.11.6 *Message Code Identifier*: The Message Code Identifier parameter will be derived from the OBU and describes the outgoing message in relation to its Group type. This is a Mandatory parameter and will be in the range 0-255. Again, a limited range of Message Codes and their corresponding meaning have been assigned and additional Codes may be included at a later date.

4.11.7 *Message Parameters Length*: The Message Parameters Length is an Optional parameter and is only required if the Message Type and associated Message Code require additional qualification. The Message Parameters Length field identifies the length of the corresponding Message Parameters Data field.

4.11.8 *Message Parameters Data*: The Message Parameters Data field is an Optional parameter and may be included in the outgoing message if the Message Type and associated Message Code require additional qualification. This is a 'Free Format' field with a minimum Data Length of 1 byte and a maximum Data Length of 255 bytes.

Manual Messages

4.11.9 The following table details the structure and definition of the manually generated Outgoing Messages. The currently identified messages (Message Type and Message Code) are grouped by category and the table includes details of the optional supplementary data elements where applicable.

Message Type	Message Code	Event	Message Parameters
Emergency Messages		Highest Priority Messages	
0	0	Need Assistance	Code - Physical Assault
			Code – Passenger Sick
			Code - *Other or None*
0	1	Accident	Code - Need Emergency Services
			Code - *Other or None*
0	2	Obstruction Need to Divert	Code - *Reason or None*
0	3	Diverting	Code - *Reason or None*
0	4	Abandoning Journey	Code - *Reason or None*
0	5	Curtailing Journey	Code - *Reason or None*
0	6 … 255	Not used	Not used
Service Messages			
1	0	Request PMR Radio Session	Driver or Vehicle or Radio ID
1	1	Accept New Duty	Duty Number
1	2	Unable to Accept New Duty	Duty Number
1	3	Accept Rest Day	N/A
1	4	Unable to Accept Rest Day	N/A
1	5	Accept Overtime	N/A
1	6	Unable to Accept Overtime	N/A
1	7	Request Relief	N/A

Message Type	Message Code	Event	Message Parameters
1	8	Acknowledge Receipt of Incoming Message	N/A
1	9-255	Not used	Not used
Vehicle Status			
2	0	Puncture	N/A
2	1	Low Oil Pressure	N/A
2	2	High Engine Temp	N/A
2	3	Passenger Load It is proposed that this either be manually generated or sent when the state changes if the ETM is able to determine passenger load.	U Defines the Passenger Load: 0 = Empty 1 = ¼ 2 = ½ 3 = ¾ 4 = Full 5 = Overloaded 0xFF = Unknown
2	4-255	Not used	Not used
Unallocated			
3-126	-	Not used	Not used
Free Format Text Messages		Lowest Priority Messages	
127	0	Text String #1	C…. e.g. ‘Text String’
127	1	Predefined message	U Message number selected by the driver
127	2-255	Not used	Not used

4.11.10 *Free Format Text Messages*: The ability to support Free Format Text Messaging has been included in the above table for consideration. Depending on the type of on-board unit, it may be possible to generate outgoing text messages from the machine's User Interface.

Automatic Messages

4.11.11 The following table details the structure and definition of the automatic generated Messages. Again, the currently identified messages (Message Type and Message Code) are grouped by category and the table includes details of the optional supplementary data elements where applicable.

Message Type	Message Code	Event	Optional Parameters
Vehicle Location		**Highest Priority Messages**	
128	0	Departing from Stop	Stop ID: *CCCCCCCCCCCC* *NaPTAN Format*
128	1	Crossing TLP Trigger Line	Scheme format XXXXXXXXXXXXX encoded as either Bytes or Integers to simplify generation however values remain limited by compressed format. Functionality for Values outside the defined range is not defined. RRRR RRRR SSSS SSSS SSSS SSSS TTTT TTTT MMMM MMMM DDDD DDDD PPPP PPPP R: Region to which the Traffic Signal belongs (Value mapped by central system so no need for global coordination) S:Traffic Signal ID [0 to 16383] T:Trigger Point [0 to 9 coded as per RTIG T031 v1.2 and RTIG T008 v1.6] M:Movement No [0 to 31] D:Timetable Deviation [0 to 31 coded as per RTIGT031 v1.1] P:Priority [0 to 6 coded as per RTIGT031 v1.1] LVCC and VN not sent as control centre should know what vehicle

Commented [T1]: This is different to T031 which allows for a deviation of 0-31. Making T008 match T031 would be a significant change because of increasing length of data packets.

Commented [T2]: This is different to T031 which allows for a priority of 0-6. Making T008 match T031 would be a significant change because of increasing length of data packets.

Message Type	Message Code	Event	Optional Parameters
			this is.
128	2	Arriving Stop	Stop ID: *CCCCCCCCCCCCC* *NaPTAN Format*
128	3	Off Route. The OBU has determined that it is no longer on the route. The mechanism for determining this is up to the OBU	
128	4	On Route Event. The OBU has determined that it has returned to the route	Stop ID: *CCCCCCCCCCCCC* *NaPTAN Format*
128	5	On Diversion Stop. Message sent if Opening Doors when on Diversion (that is Off Route) to inform control centre the vehicle has stopped but stop is unknown	

Message Type	Message Code	Event	Optional Parameters
128	6	Depot Exit / Entry Depot Entry: Sent on entry to a depot “zone” as defined in the data supply of the OBU. Depot Exit: Sent on Exit from depot zone as defined in OBU data supply On Start Up if this message is supported Depot Entry should be sent if OBU is located within a depot zone.	Entry / Exit Flag: X 0 = Depot Exit 1 = Depot Entry
128	7-255	Not used	Not used
Vehicle Status			
129	0	Vehicle Configuration Information Sent in response to a request for config information If not supported	CCC…… in following format: [AAAA:BB.CC.ZZ:DD/MM/YY] AAAA Code to identify the object BB - Major Version CC - Sub Version ZZ - Minor Version DD - Day From Which Active MM - Month From Which Active YY - Year From Which Active

Message Type	Message Code	Event	Optional Parameters
		then error code can be sent with NACK	Field can repeat with another set of [….] using [and] as the delimiters. Should be sent for all configuration information. The meaning of AAAA is OBU specific however the following are proposed OBOS – OBU Operating System OBFM – OBU Firmware OBAP – OBU Application OBDS – OBU Data Supply OBFI – OBU Fare Information RDOS – Radio OS RDFM – Radio Firmware NSFM – Next Stop Firmware
129	1	Vehicle Serial No Info Sent in response to a request for Serial No information If not supported then error code can be sent with NACK	CCC……in following format: [AAAA:NNNNNNNNNNN] AAAA Code to identify the device NNNNNNNNNNNNN....NN Serial Number of device N is AlphaNumeric. Send this code multiple times or with multiple entries delimited by [....]
129	2-255	Not used	Not used
Unallocated			
130-199	-	Not used	Not used
200-239	-	To be used by for proprietary codes without control	

Message Type	Message Code	Event	Optional Parameters
240-249	-	To be used for testing and development of new codes	This area is intended for implementation of new codes which will be published. Therefore access to this area is only via RTIG control – A test code will be allocated and may be used until such point as the code is agreed and then it will be published into the appropriate section of the specification
250-254	0-255	To be uniquely assigned pairs of codes for supplier specific extensions	The allocation of codes is to be managed by RTIG. A supplier should register for the allocation of a code pair and this will be recorded as being used by that supplier. The meaning of the code is not published and therefore will only be supported by the registering supplier. **It is strongly recommended that this is not used for mission critical data exchange – a new feature can be added by defining a published extension to this specification.**
Unallocated		**Lowest Priority Messages**	
255	0	TBA	TBA
255	255	TBA	TBA

4.12 Event (centre to OBU) – Message ID #60

This Message Type does require an acknowledgement

Please note that this Message Type is likely to be the subject of considerable further development.

Parameter	Mandatory/ Optional	Null Value Allowed?	Format	Length (bytes)
Message ID	Mandatory	No	X	1
Sequence ID	Mandatory	No	XX	2
Reference Sequence ID	Mandatory	No	XX	2
Message Type	Mandatory	No	X *See table below*	1
Message Code	Mandatory	No	X *See table below*	1
Message Parameters Length	Optional	No	X *See table below*	1
Message Parameters Data	Optional	No	Qualifying Data	1-255

4.12.1 *Message ID*: This parameter uniquely defines the message type. The Message ID requires a single byte capable of describing 255 (1-255) message types with 0 (Zero) being reserved.

4.12.2 *Sequence ID*: The Sequence ID will be under the control of the communications centre and will provide a unique reference for each event message sent to the OBU. This is a Mandatory parameter and shall be in the range 1-65535 with 0 (Zero) being reserved.

4.12.3 *Reference Sequence ID*: The Reference Sequence ID may be used by the communications centre in order to reference a previous Message from the OBU. This reference will be used in subsequent messages if communications centres wish to make further enquiries regarding a specific event. This is a Mandatory parameter and will be in the range 1-65535. If the reference to a previous Outgoing Message is not required then this field will be set to 0 (Zero).

4.12.4 *Message Type Identifier*: The Message Type Identifier parameter will be under the control of the communications centre and describes the Group type of the incoming message. At present, a limited range of Group types have been assigned and additional Groups may be included at a later date. This is a Mandatory parameter and will be in the range 0-255.

4.12.5 *Message Code Identifier*: The Message Code Identifier parameter will be under the control of the communications centre and describes the incoming message in relation to its Group type. This is a Mandatory parameter and will be in the range 0-255. Again, a limited range of Message Codes and their corresponding meaning have been assigned and additional Codes may be included at a later date.

4.12.6 *Message Parameters Length*: The Message Parameters Length is an Optional parameter and is only required if the Message Type and associated Message Code require additional qualification. The Message Parameters Length field identifies the length of the corresponding Message Parameters Data field.

4.12.7 *Message Parameters Data*: The Message Parameters Data field is an Optional parameter and may be included in the incoming message if the Message Type and associated Message Code require additional qualification. This is a 'Free Format' field with a minimum Data Length of 1 byte and a maximum Data Length of 255 bytes.

Communication Centre Messages

4.12.8 The following table details the structure and definition of the communications centre generated Messages. The currently identified messages (Message Type and Message Code) are grouped by category and the table includes details of the optional supplementary data elements where applicable.

Message Type	Message Code	Event	Message Parameters
Emergency Messages		**Highest Priority Messages**	
0	0	Acknowledge Receipt of all Emergency Messages	Code - *Other or None*
0	1-254	Not used	Not used
0	255	TBA	---
Service Message			

Message Type	Message Code	Event	Message Parameters
1	0	Acknowledge Receipt of Outgoing Message	N/A
1	1	Request PMR Radio Session	Driver or Vehicle or Radio ID
1	2	Start New Duty Request	Duty Number
1	3	Start Rest Day Request	Date & Time
1	4	Overtime Request	N/A
1	5	Advise Accident	Location Reference *NaPTAN Format*
1	6	Advise Diversion	Location Reference *NaPTAN Format*
1	7	Advise Service Termination	Time or Stage or Location
1	8	Advise Crew Change	Driver Number/Name
1	9	Grant Relief	Date & Time
1	10	Deny Relief	N/A
1	11	Set Journey On receiving this message the OBU should attempt to log into this Running Board and determine (where it has a data supply) a suitable journey based on current location and time of day. The OBU should always send a Journey Details (basic or advanced) message to the control centre (see 4.74.6) to inform it of the change	CCCCCC – Service Code CCCCCCC – Running board* CCCCC – Journey Number* hhmm – Scheduled Start CCCCCC – Duty Number* CCCCCC – Public Service Code X – Direction *Where data is not available a field of valid nulls shall be used.* *Reference sections 4.6 and 4.7 Journey Details message for information on each of the fields.* *'*' =Operator defined*

Message Type	Message Code	Event	Message Parameters
1	12	Request Config Info Message sent to ask the OBU to serialise the configuration information to the server	N/A
1	13	Request Serial No Message sent to ask the OBU to send serial no information.	N/A
1	14-255	Not used	Not used
Vehicle Status			
2	0	Acknowledge Receipt of Outgoing Message	N/A
2	1	Send Schedule Deviation to the Vehicle	S Schedule Deviation in multiples of 30 seconds 0 – 0x7F positive 0xFF – 0x81 negative 0x80 indicate unknown and schedule deviation should not be shown until a valid update is received.
2	2-255	TBA	TBA
Notification			
3	**0**	Error Notification	X Error number as per acknowledgement structure
Unallocated			
4-126	-	TBA	TBA
Free Format Text Messages		**Lowest Priority Messages**	
127	0	Acknowledge Receipt of Outgoing Message	N/A

Message Type	Message Code	Event	Message Parameters
127	1	Text String #1	XC…. X = Destination (Bit 1 = Driver, Bit 2 = Passengers, Bit 3 = Destination Display, 4 = Audio (text to speech)) as a binary pattern 'Text String'
127	2	Predefined message	XX First Byte: Message from data supply to be displayed. Second: Destination (Bit 1 = Driver, Bit 2 = Passengers, Bit 3 = Destination Display, 4 = Audio (text to speech)) as a binary pattern
127	3-254	Not used	Not used
127	255	TBA	TBA
Unallocated			
128-199	-	Not used	Not used
129	1	Text Message to display to Driver with alert tone	C… Display text message to driver immediately without requiring any driver interaction.
200-239	-	To be used by for proprietary codes without control	
240-249	-	To be used for testing and development of new codes	This area is intended for implementation of new codes which will be published. Therefore access to this area is only via RTIG control – A test code will be allocated and may be used until such point as the code is agreed and then it will be published into the appropriate section of the specification

Message Type	Message Code	Event	Message Parameters
250-254	0-255	To be uniquely assigned pairs of codes for supplier specific extensions	The allocation of codes is to be managed by RTIG. A supplier should register for the allocation of a code pair and this will be recorded as being used by that supplier. The meaning of the code is not published and therefore will only be supported by the registering supplier. **It is strongly recommended that this is not used for mission critical data exchange – a new feature can be added by defining a published extension to this specification.**

4.13 Enquiry – Message ID #255

This Message Type does require an acknowledgement

4.13.1 The Enquiry Message generated by the Server may used to determine the availability of any given vehicle within the scheme. This message is unsolicited and may be generated by the Server at any time during a session.

Parameter	Mandatory/ Optional	Null Value Allowed?	Format	Length (bytes)
Message ID	Mandatory	No	X	1
Session Vehicle Identifier	Mandatory	No	XX	2

4.13.2 *Message ID*: This parameter uniquely defines the message type. The Message ID requires a single byte capable of describing 255 (1-255) message types with 0 (Zero) being reserved.

4.13.3 *Session Vehicle Identifier*: The Session Vehicle Identifier parameter uniquely identifies the specific vehicle. This parameter requires two bytes and will be in the range 1-65535.

5 Communications management

5.1 Server Reboot

5.1.1 Upon reboot, the Server is not required to retain which session ids are allocated to which OBUs. However, the Server is required to resume session id allocation following the last id allocated prior to reboot.

5.1.2 When the Server receives a message which has an unknown SVID and which requires an acknowledgement, it will respond with a NACK containing error code 1 (unknown sender/SVID).

5.1.3 If the Server receives a message which has an unknown SVID but which does not request an acknowledgement, it may send an Event (Centre to OBU) with Message Type 3 and Message Code 0 (Error Notification), indicating the error code 1 (unknown sender/SVID).

5.1.4 On receipt of such a NACK or Error Notification, the OBU must issue a new Log On Request. The Server will respond with a Log On Response which will provide a new SVID.

5.1.5 Since the server will not, after reboot, immediately re-issue session IDs which were active prior to reboot, there will be sufficient time for an OBU to detect the need for a new Log On Request before the server re-allocates the session ID to a different OBU.

5.2 Session Timeout

5.2.1 The Server expects to receive regular messages (position updates) after an OBU has logged on. If the Server does not receive any messages for a configurable period, it can time the session out. Optionally, it can issue an Enquiry message to determine whether communications with the OBU is still possible – if the Server receives an acknowledgement, it should not time the session out.

5.2.2 The OBU must implement a session timeout. This must be shorter than the period configured for the server in 5.2.1 above. This prevents the server from re-issuing the session id to a different OBU and the original OBU then resuming communications on the same session.

5.3 OBU Restart

5.3.1 If the OBU restarts, it must issue a new Log On Request. It cannot assume that any previously issued SVID is still valid.

5.3.2 If the Server has an existing SVID assigned to the OBU which has not yet timed out, it will return this SVID in a Log On Response, otherwise a new SVID will be allocated.

5.4 Failure of communications link

5.4.1 It is recommended that the OBU set the ACK required flag in a Position Update message at a regular interval, allowing it to detect when the comms link has failed. The interval will depend on such factors as frequency of position updates, size of known radio black spots and minimum update requirements. (Due to concerns over the amount of data, the original position update message was not specified to require an ACK. However, experience in the pilot suggests that there are insufficient other messages requiring ACKs and an OBU may not detect when action is required to re-establish the link.)

5.4.2 The OBU should determine the minimum action required to re-establish DAIP level communications.

5.4.3 After a temporary loss (e.g. due to a GPRS black spot), the OBU will resume sending messages and receiving acknowledgements. If an ACK is received, the session has not been timed out by the server and the OBU may proceed with its current SVID. If a NACK with error code 1 (unknown sender/SVID) is received, the OBU is required to send a Log On Request.

5.4.4 When messages are not acknowledged for a more extended period, it is recommended that the OBU perform a low level network ping to the server IP address, to determine if the OBU remains on the network.

5.4.5 If there is no response to the ping, it is recommended that the OBU deconstruct its network connection, re-establish the network context from the beginning then attempt to contact the server again.

5.4.6 The OBU needs to disconnect from the network as completely as possible prior to fully rebooting the GSM module. This is to ensure that the network context throughout the GPRS networking layer is reset.

5.4.7 The GPRS modem should be forced to detach from the network both in GSM and GPRS mode. The modem should then be reset and forced to do a full cold restart. This should ensure that the connection to the network is fully re-established.

5.4.8 If the GPRS modem is simply reset, performing a warm start because it thinks it already has the network details, this can result in the modem and the network getting into an incompatible state; the modem will insist it is connected but no data can flow. Just attempting to reinitialise the GPRS connection will not return the link to operational.

5.4.9 Once the communications link has been re-established, the OBU must issue a Log On Request.

6 Data Supply Extension

6.1 Introduction

6.1.1 This extension is used to define file formats which can be used by the control centre system to provide information to the external OBU management system for download, where supported, to the OBU. This is not a mandatory extension however it has been introduced in an effort to standardise some aspects where a shared data supply would be greatly beneficial.

6.1.2 On the first line of the data file is a version information header. This should be sent to the vehicle and can be requested using the Get Configuration command.

6.2 Bus Priority Data Supply

6.2.1 The specification includes a mechanism to send bus priority telegrams to the control centre, the definition of locations at which messages are triggered is contained within RTIGT042 Traffic Light Priority Trigger Position File Format.

6.2.2 RTIGT042 is an XML based format and supersedes the initial extension in versions up to v1,2 which used a very simple comma separated variable format which is described here for completeness.

6.2.3 The CSV file is structured with a first line as version number, second line as a header and then data to define each trigger point containing the following fields:

- Identifier – unique number assigned to each record. Only used for identification
- Service Code – the Service the vehicle needs to be on to trigger this request
- Direction – Direction of the service
- Longitude location as number WGS84 coordinate
- Latitude location as number WGS84 coordinate
- Region ID
- Traffic Signal ID
- Trigger point
- Movement number
- Capture zone diameter – how close to the point the vehicle needs to be (in Meters)

6.3 Driver and Control Centre Messages Data Supply

6.3.1 In V1.1 of the specification a mechanism for displaying predefined messages is introduced. These messages are system specific however a standardised format for passing these messages is defined in this extension.

6.3.2 It uses a comma separated variable format with a first line as version number, second line as a header and then data to define each message containing the following fields:

- Message ID
- Message Text in quotes “” to define the message to be sent

7 Protocol management and extensions

7.1 Introduction

7.1.1 This section describes the approach to be taken to manage this specification, resolve unclear issues with the specification and coordinate extensions to the specification.

7.1.2 RTIG maintains a standing Working Group responsible for managing this specification which will meet at regular intervals, to be determined by the working group, and review any issues, clarifications or extensions that have been raised. Such meetings may be physical, conference call or email circulation as is deemed appropriate.

7.1.3 The Working Group should look to minimise overlapping extensions, where possible any proposed extension should be extended if required to cover the range of functionality that is likely to be required.

7.1.4 The Working Group shall allocate a new version number to the specification on each issue of the specification.

7.2 Clarification

7.2.1 Where an issue is identified by parties involved in the implementation of aspects of this specification, this issue should be sent to the RTIG Secretariat (secretariat@rtig.org.uk) where the issue will be recorded. It will be circulated to the standing Working Group who will either convene a meeting or add it to the agenda of the next meeting.

7.3 New Feature or Feature extension

7.3.1 Any new feature which is felt should be added to the specification can be proposed to the RTIG Secretariat (secretariat@rtig.org.uk) where the issue will be recorded. Where possible the proposal should contain as much information as possible, including the structure of any required message, any implementation details and interaction required. This will then be circulated to the standing Working Group who will review the proposal and take the appropriate action: include in the next revision as is or with modification or reject the request with an explanation or a request for more information.

7.3.2 The updated specification will be sent to the originators of new features as well as the work group for review prior to issuing.

7.3.3 The above procedure should be followed even where the extension is proposed by a member of the Working Group.

A Example message sequences (informative)

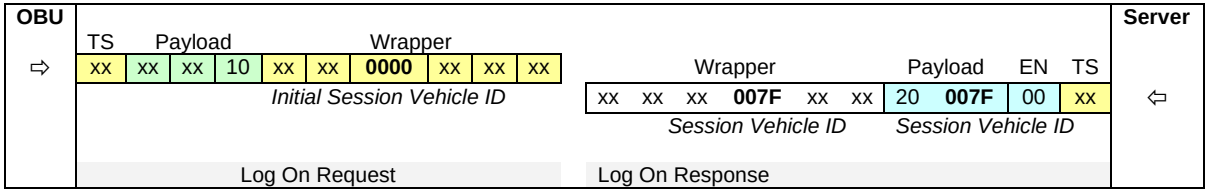
A.1 Introduction

A.1.1 The following message sequence diagrams indicate the communication process between the OBU and Back Office Server. The messages are typical and where appropriate, highlight the key data fields within each message type.

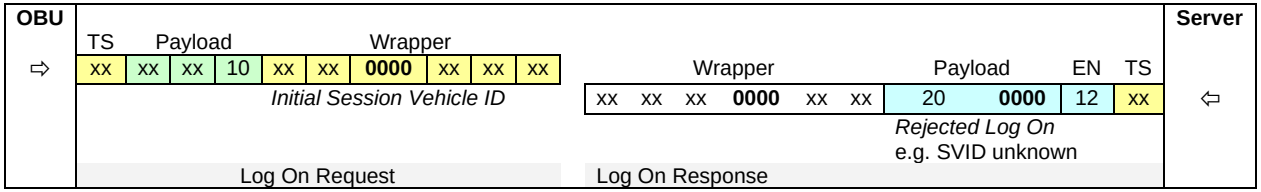
A.1.2 For clarity, the diagrams omit the UDP element of the telegram.

A.2 Session Log On Sequences

Successful Log On without Communication Failures



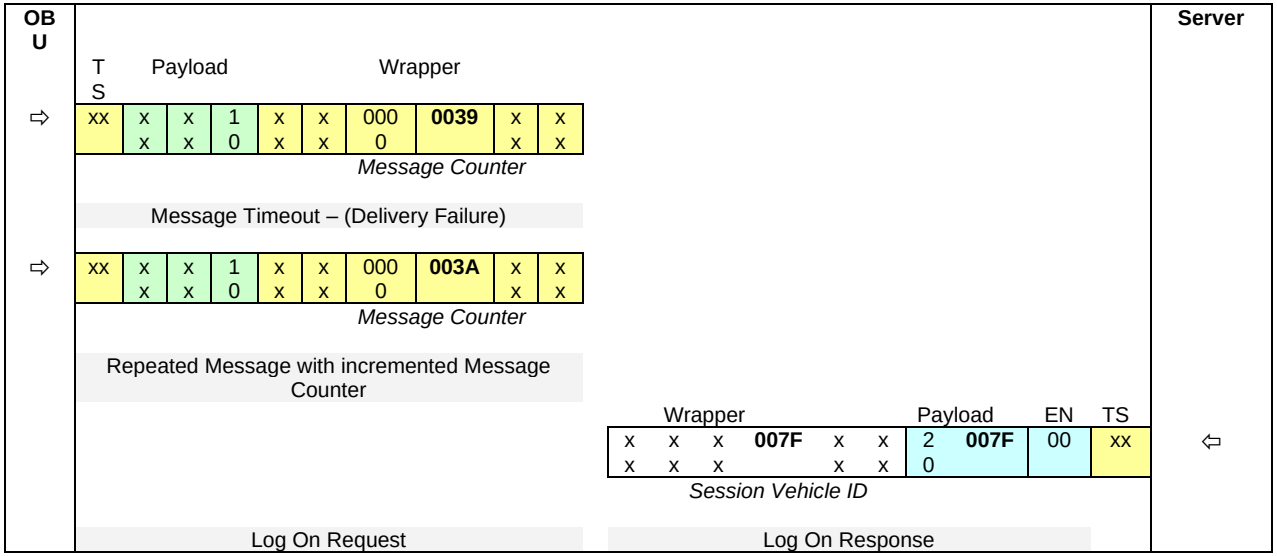
Failed Log On without Communication Failures



Notes:

- For the initial Session Log On request, the OBU will use the default Session Vehicle ID 0000 Hex.
- Upon successful receipt of a valid Log On request, the Server will issue a unique ID for the current session, e.g. 007F Hex
- Failure to Log On should not prevent the OBU from entering operational duty.

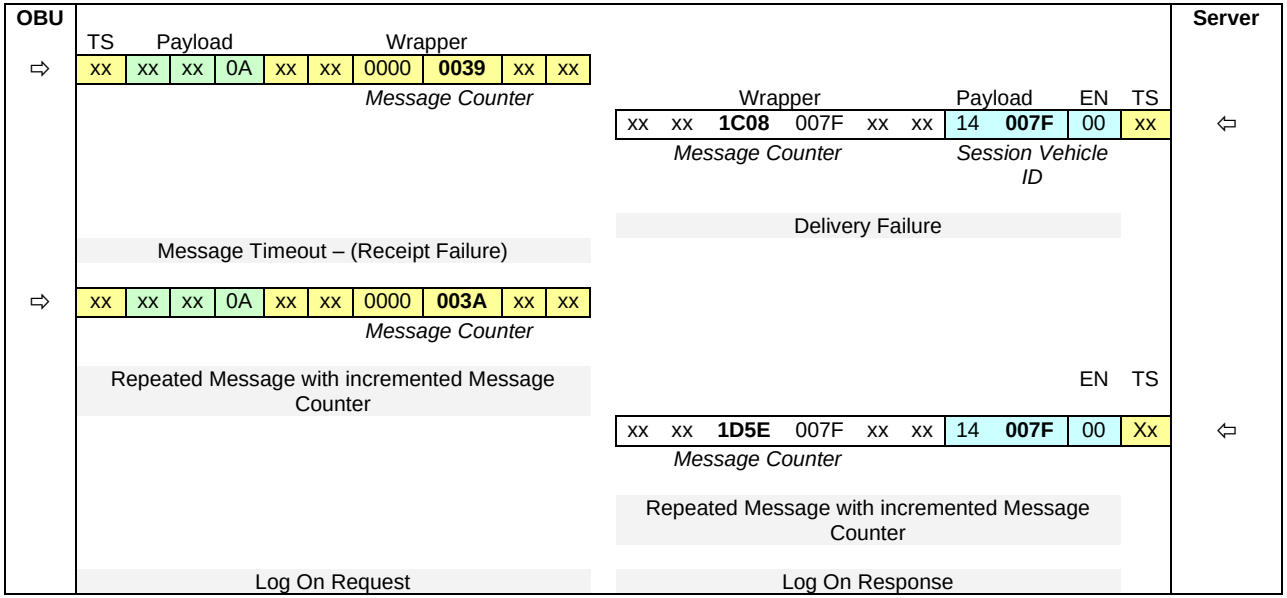
Successful Log On with OBU Delivery Failure



Notes:

- In this example, the initial Log On message, issued by the OBU does not reach the Server. After a pre-determined Timeout, the OBU re-transmits but with an incremented Message Count; 003A Hex.
- After a suitable number of retry attempts, the OBU will cease transmitting and audit the failed communication session.
- Failure to Log On should not prevent the OBU from entering operational duty.

Successful Log On with Server Delivery Failure

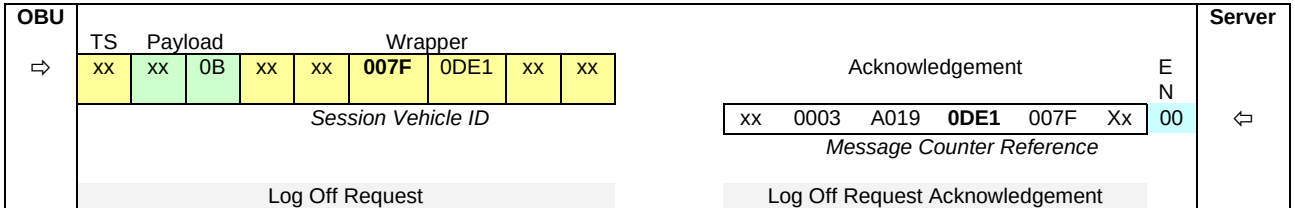


Notes:

- Message IDs are here shown for consistency in Hex (Log On Request 0A = 10(dec), etc)
- In this example, the initial Log On response, issued by the Server does not reach the OBU. After a pre-determined Timeout, the Server re-transmits but with an incremented Message Count; 1D5E Hex.
- After a suitable number of retry attempts, the OBU will cease transmitting and audit the failed communication session.
- Failure to Log On should not prevent the OBU from entering operational duty.

A.3 Session Log Off Sequences

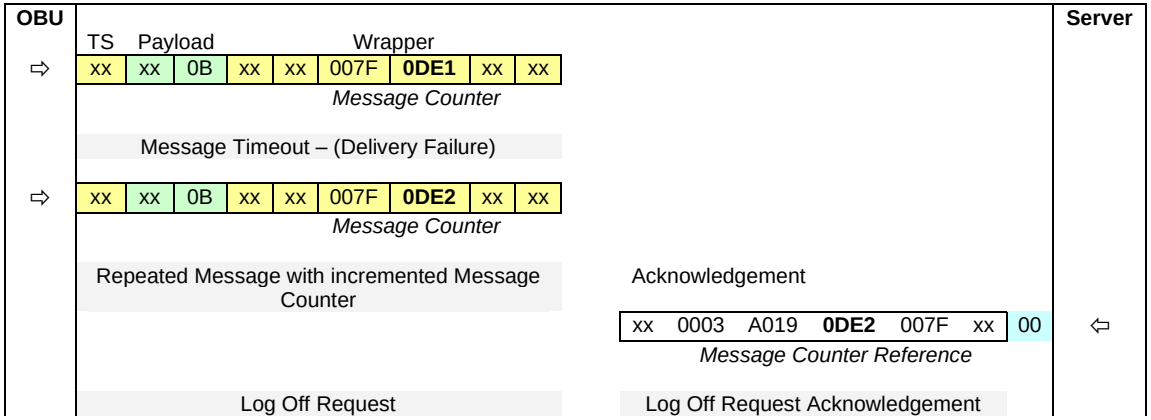
Successful Log Off without Communication Failures



Notes:

- In this example, the Server acknowledges the Log Off message, issued by the OBU.

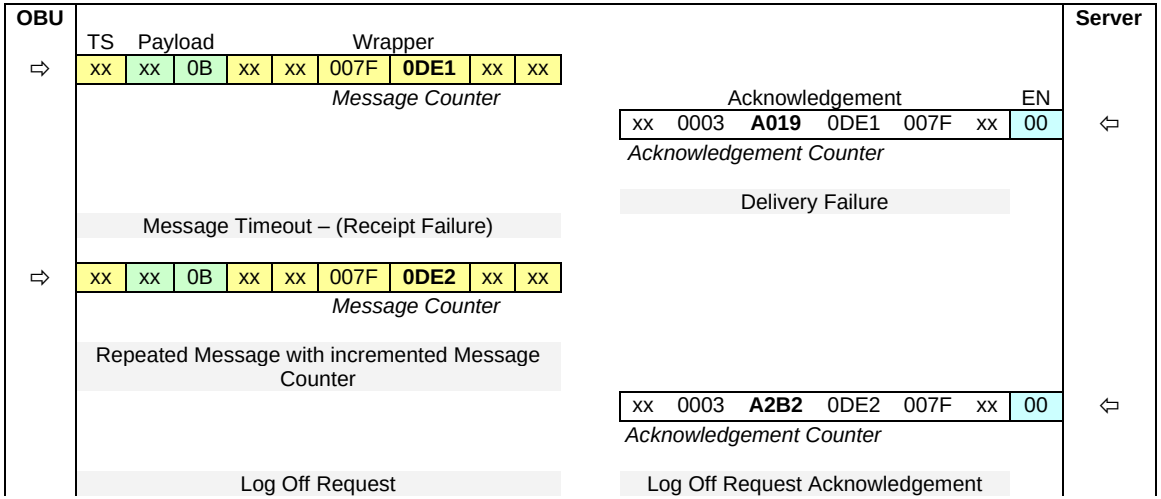
Successful Log Off with OBU Delivery Failure



Notes:

- In this example, the Log Off message, issued by the OBU does not reach the Server. After a pre-determined Timeout, the OBU re-transmits but with an incremented Message Count; 0DE2 Hex.
- After a suitable number of retry attempts, the OBU will cease transmitting and audit the failed communication session.
- Failure to receive the Log Off acknowledgement should not prevent the OBU from clearing down and the subsequent Log On sequence will again use the default Session Vehicle ID 0000 Hex. Upon receipt of the new Log On message, the previous Session Vehicle ID, allocated by the Server will be discarded and a new ID allocated.

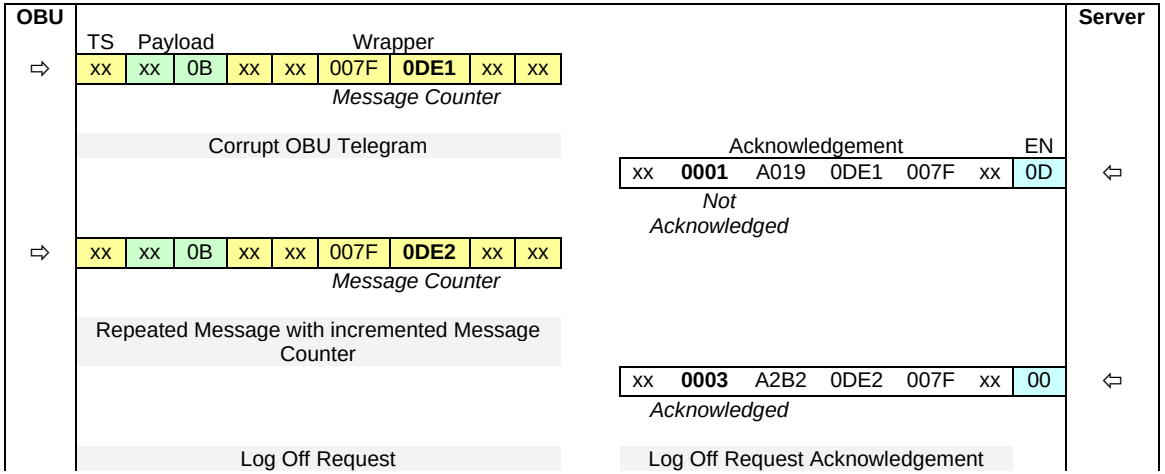
Successful Log Off with Server Delivery Failure



Notes:

- In this example, the Log Off message acknowledgement, issued by the Server does not reach the OBU. After a pre-determined Timeout, the OBU re-transmits but with an incremented Message Count; 0DE2 Hex.
- After a suitable number of retry attempts, the OBU will cease transmitting and audit the failed communication session.
- Again, failure to receive the Log Off acknowledgement should not prevent the OBU from clearing down and the subsequent Log On sequence will again use the default Session Vehicle ID 0000 Hex. Upon receipt of the new Log On message, the previous Session Vehicle ID, allocated by the Server will be discarded and a new ID allocated.

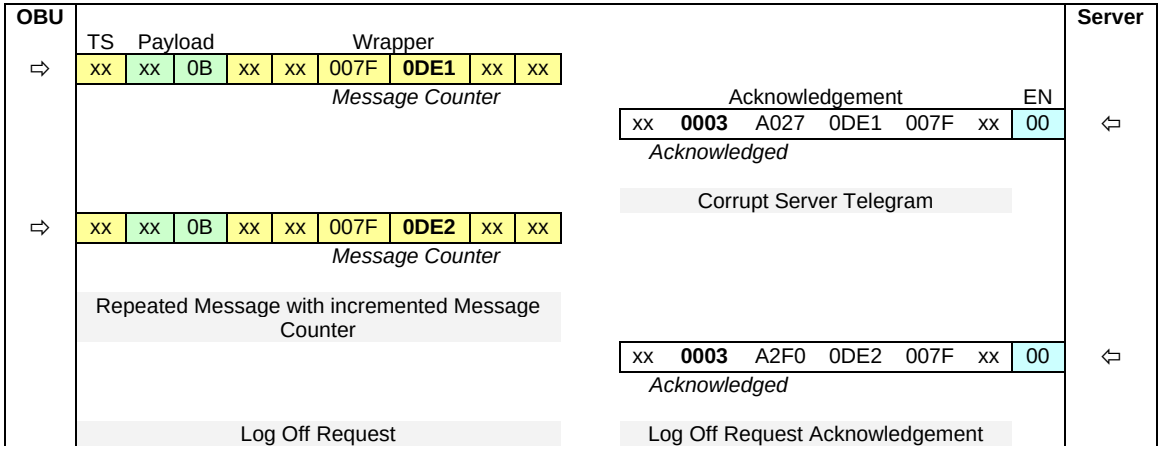
Successful Log Off with Corrupt OBU Telegram



Notes:

- In this example, the Log Off message, issued by the OBU, is corrupted upon receipt. In response, the Server issues a Negative Acknowledgement, NAK and upon receipt, the OBU re-transmits but with an incremented Message Count; 0DE2 Hex.
- After a suitable number of retry attempts, the OBU will cease transmitting and audit the failed communication session.
- Again, failure to receive the Log Off acknowledgement should not prevent the OBU from clearing down and the subsequent Log On sequence will again use the default Session Vehicle ID 0000 Hex. Upon receipt of the new Log On message, the previous Session Vehicle ID, allocated by the Server will be discarded and a new ID allocated.
- If after 5 attempts, the OBU has not received an Acknowledgement, it will cease transmitting and audit the failed communication session.

Successful Log Off with Corrupt Server Telegram

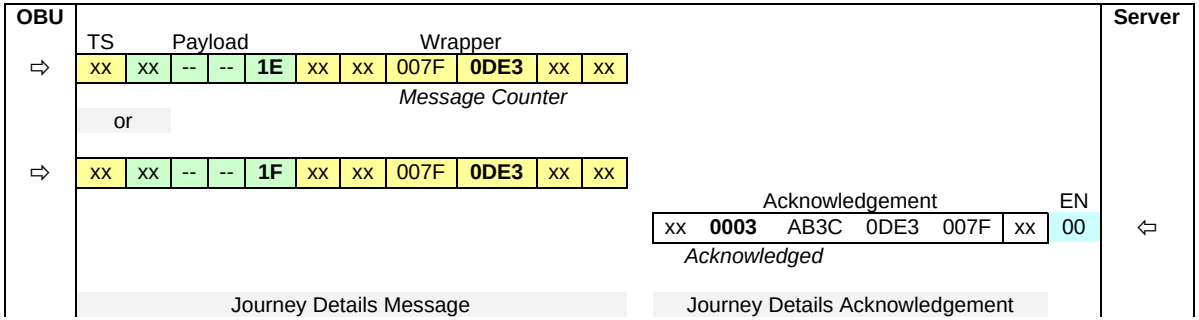


Notes:

- For this example, the Log Off message, issued by the Server, is corrupted upon receipt by the OBU. In response, the OBU re-transmits but with an incremented Message Count; 0DE2 Hex.
- After a suitable number of retry attempts, the OBU will cease transmitting and audit the failed communication session.
- Again, failure to receive the Log Off acknowledgement should not prevent the OBU from clearing down and the subsequent Log On sequence will again use the default Session Vehicle ID 0000 Hex. Upon receipt of the new Log On message, the previous Session Vehicle ID, allocated by the Server will be discarded and a new ID allocated.
- If after 5 attempts, the OBU has not received an Acknowledgement, it will cease transmitting and audit the failed communication session.

A.4 Journey Details Sequences

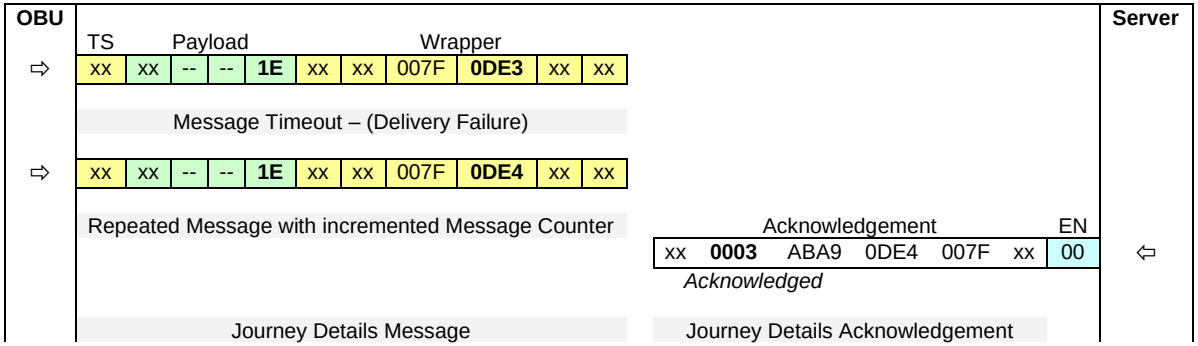
Successfully Acknowledged Journey Details without Communication Failures



Notes:

- In the example above, the Journey Details message, issued by the OBU, may take one of two variants; e.g. Message ID #30(dec) (1E in Hex) or #31(dec) (1F in Hex). In response, the Server will issue an Acknowledgement upon receipt.

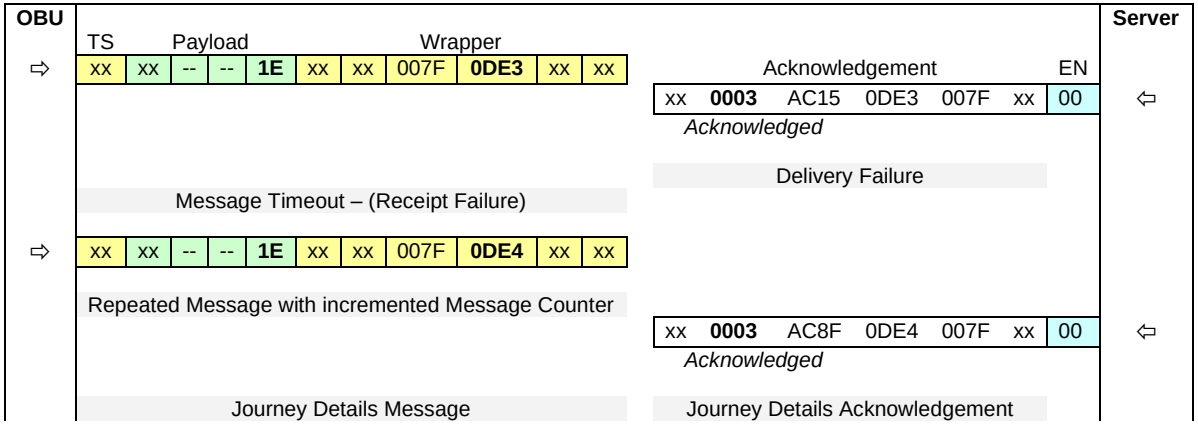
Successfully Acknowledged Journey Details with OBU Delivery Failure



Notes:

- In the example above, the Journey Details message, issued by the OBU, does not reach the Server. After a pre-determined Timeout, the OBU re-transmits but with an incremented Message Count; 0DE4 Hex.
- In response, the Server will issue an Acknowledgement upon receipt.
- If after 5 (five) attempts, the OBU has not received an Acknowledgement, it will cease transmitting and audit the failed communication session.
- Failure to receive an Acknowledgement should not prevent the OBU from entering operational Service.

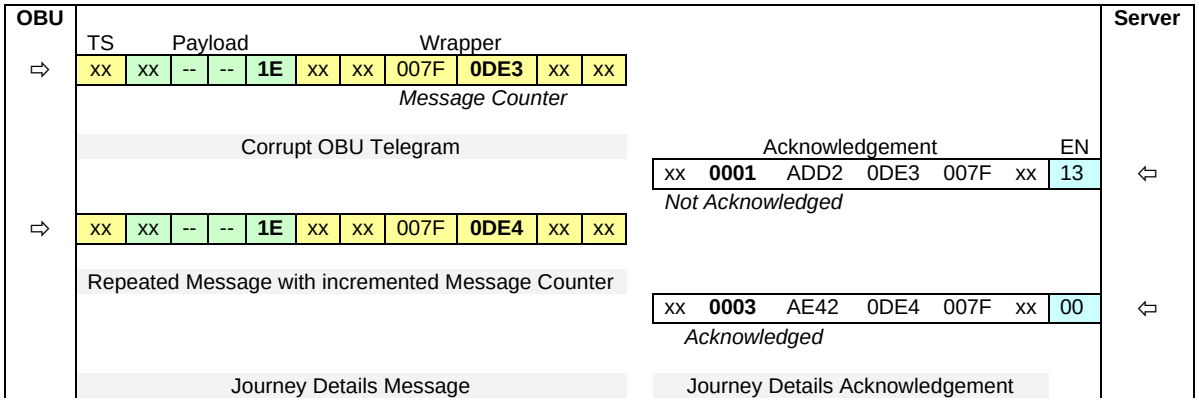
Successfully Acknowledged Journey Details with Server Delivery Failure



Notes:

- In the example above, the Journey Details Acknowledgement message, issued by the Server, does not reach the OBU. After a pre-determined Timeout, the OBU re-transmits but with an incremented Message Count; 0DE4 Hex.
- In response, the Server will again issue an Acknowledgement upon receipt.
- If after 5 (five) attempts, the OBU has not received an Acknowledgement, it will cease transmitting and audit the failed communication session.
- Failure to receive an Acknowledgement should not prevent the OBU from entering operational Service.

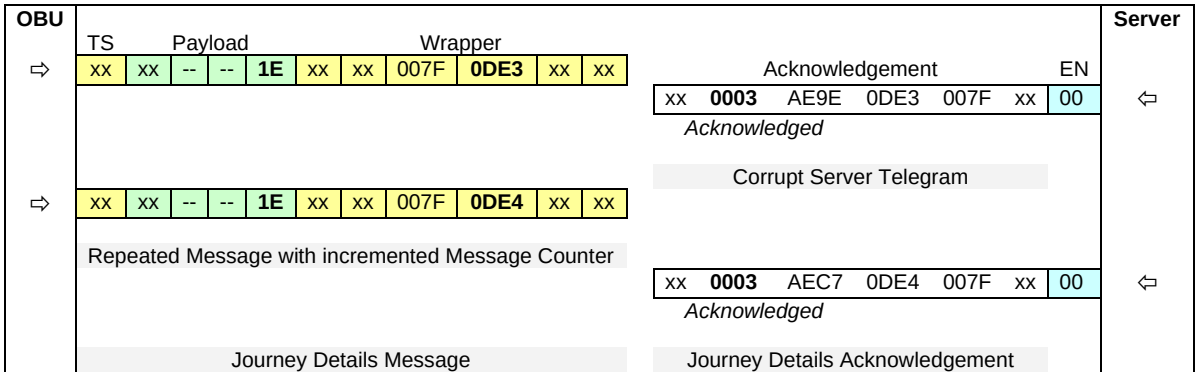
Successfully Acknowledged Journey Details with Corrupt OBU Telegram



Notes:

- In the example above, the Journey Details message, issued by the OBU, is corrupted upon receipt. In response, the Server issues a Negative Acknowledgement, NAK and upon receipt, the OBU re-transmits but with an incremented Message Count; 0DE4 Hex.
- In response, the Server will again issue an Acknowledgement upon receipt.
- If after 5 (five) attempts, the OBU has not received an Acknowledgement, it will cease transmitting and audit the failed communication session.
- Failure to receive an Acknowledgement should not prevent the OBU from entering operational Service.

Successfully Acknowledged Journey Details with Corrupt Server Telegram

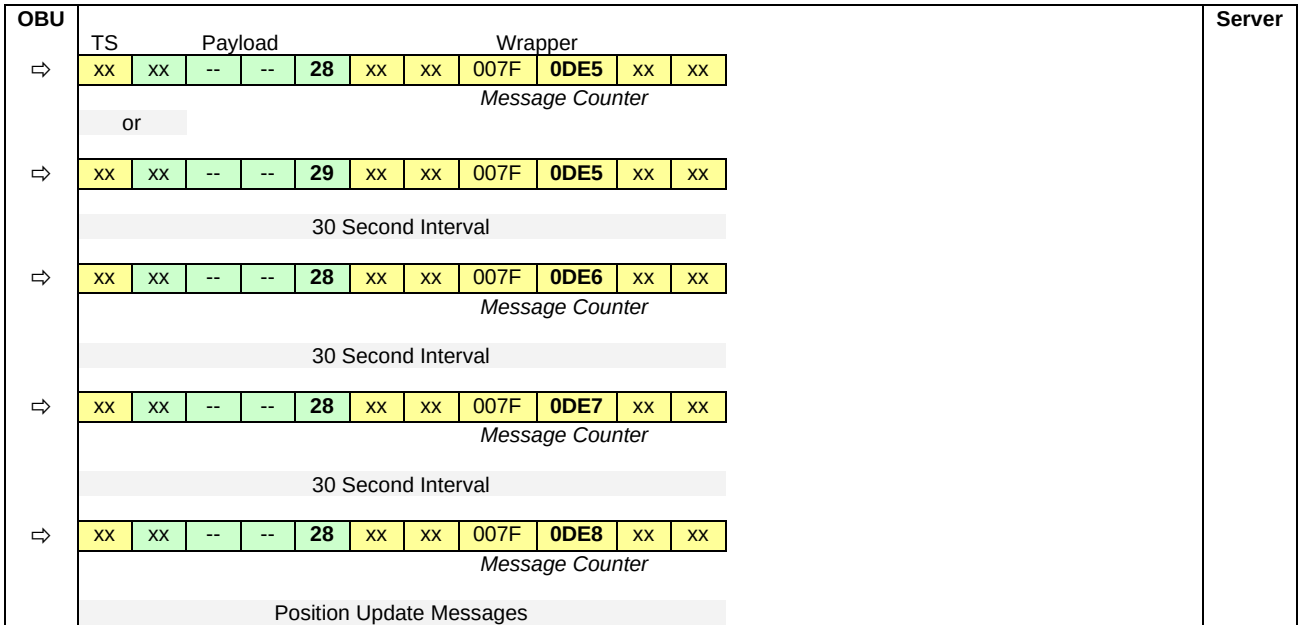


Notes:

- For this example, the Journey Details Acknowledgement message, issued by the Server, is corrupted upon receipt by the OBU. In response, the OBU re-transmits but with an incremented Message Count; 0DE4 Hex.
- In response, the Server will again issue an Acknowledgement upon receipt.
- If after 5 (five) attempts, the OBU has not received an Acknowledgement, it will cease transmitting and audit the failed communication session.
- Failure to receive an Acknowledgement should not prevent the OBU from entering operational Service.

A.5 Position Update Sequence

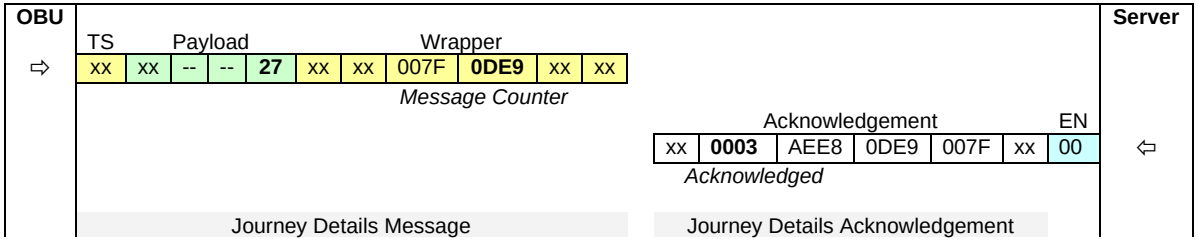
Position Update Messages



- Having entering operational Service, the OBU will automatically transmit a Positional Update message every 30 seconds.
- In the example above, the Positional Update message, issued by the OBU, may take one of many variants; e.g. Message ID #40 or #41.

A.6 End of Journey Sequence

Successfully Acknowledged End of Journey without Errors

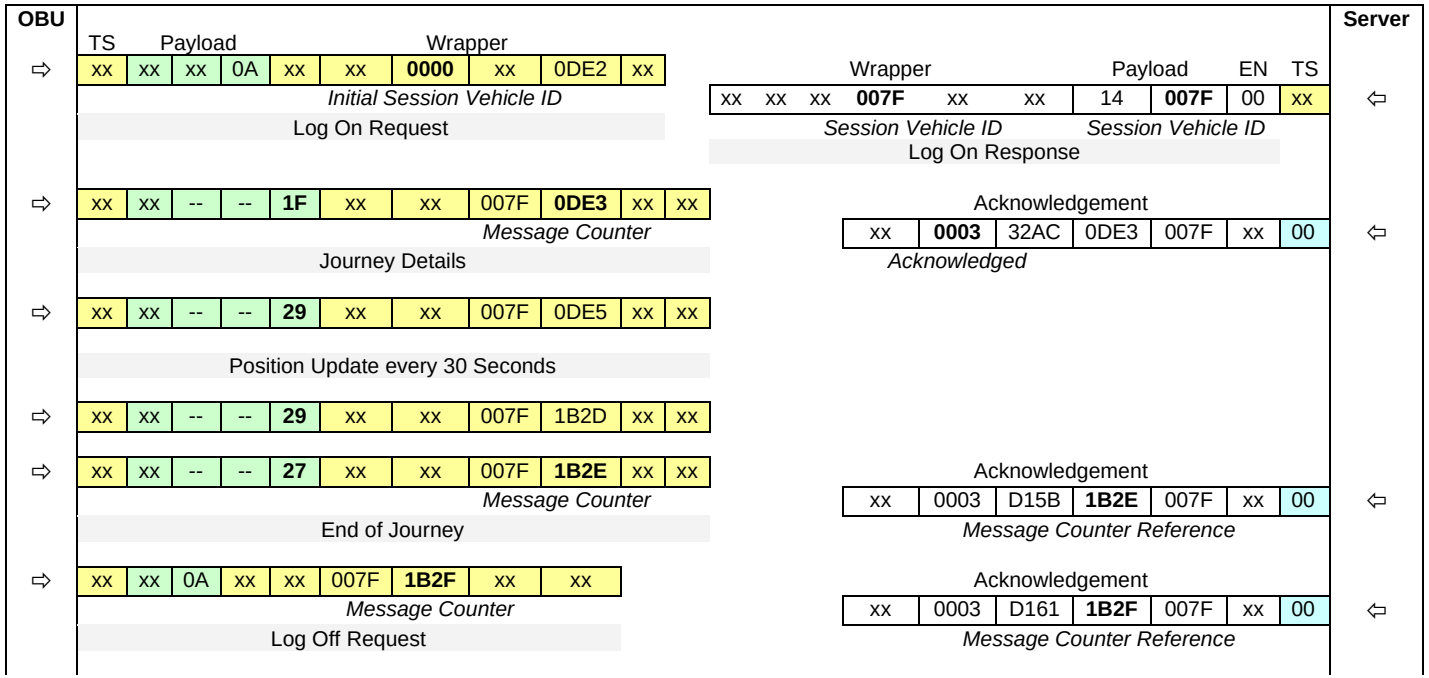


Notes:

- In the example above, the Server acknowledges the End of Journey message, issued by the OBU.
- If however, the End of Journey message does not reach the Server or the Acknowledgement message, issued by the Server, does not reach the OBU, then after a pre-determined Timeout, the OBU re-transmits but with an incremented Message Count.
- In response, the Server will again issue an Acknowledgement upon receipt.
- If after 5 (five) attempts, the OBU has not received an Acknowledgement, it will cease transmitting and audit the failed communication session.
- Failure to receive an Acknowledgement should not prevent the OBU from closing the operational Service.

A.7 Complete Sequence

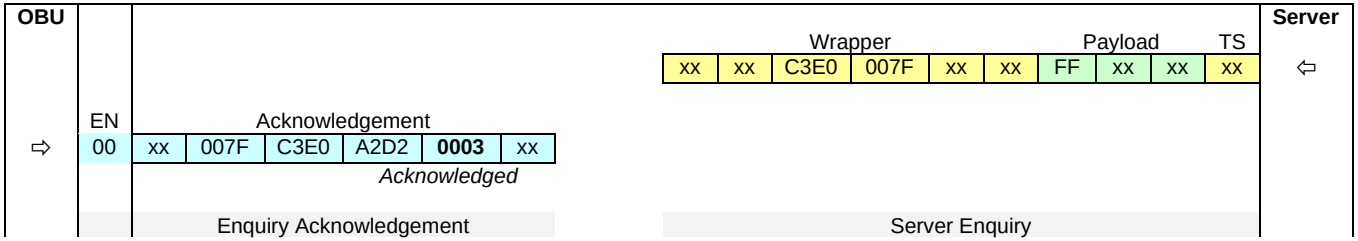
Complete Sequence without Errors



- This example offers a typical overview of an OBU/Server communication session for a single Journey.

A.8 Sever Enquiry Sequence

Server Enquiry Sequence without Errors



Notes:

- In the example above, the Server Enquiry is Acknowledged by the OBU.

B Example messages (informative)

B.1 Introduction

B.1.1 The following message examples detail the typical communication between the OBU and Back Office Server.

B.1.2 For clarity, the examples omit the UDP element of the telegram.

B.2 Session Log On Sequences

Successful Log On without Communication Failures

Log On Request from Bus

Byte Number	Data (HEX)	Parameter	Value	Wrapper/Payload
1	01	Format Version Number	1.00	Wrapper
2	00			
3	02	Message Flags	Ack Not Required	
4	04	Message Counter	1060	
5	24			
6	00	Session Vehicle ID	0	
7	00			
8	00	Optional Data Fields	0	
9	00			
10	0A	Message ID	10	Payload
11	50	Operator ID	PB35216	
12	42			
13	33			
14	35			
15	32			
16	31			
17	36			

Byte Number	Data (HEX)	Parameter	Value	Wrapper/Payload
18	00			
19	00			
20	59	Vehicle ID	YD 55 YWD	
21	44			
22	35			
23	35			
24	59			
25	57			
26	44			
Optional OBU ID Length and OBU ID may be inserted at this location				
27	09	Message Timestamp	090616124030	Wrapper
28	06			
29	16			
30	12			
31	40			
32	30			

Log On Response from Server

Byte Number	Data (HEX)	Parameter	Value	Wrapper/Payload
1	01	Format Version Number	1.00	Wrapper
2	00			
3	03	Message Flags	Ack Not Required	
4	B4	Message Counter	46146	
5	42			
6	00	Session Vehicle ID	0	
7	24			
8	00	Optional Data Fields	0	
9	7F			

Byte Number	Data (HEX)	Parameter	Value	Wrapper/Payload
10	14	Message ID	20	Payload
11	00	Session Vehicle Identifier	127	
12	7F			
13	00	Error Number	00	
Optional Server Address and Server Port may be inserted at this location				
14	09	Message Timestamp	090616124035	Wrapper
15	06			
16	16			
17	12			
18	40			
19	35			

B.3 Session Log Off Sequences

Successful Log Off without Communication Failures

Log Off Request From Bus

Byte Number	Data (HEX)	Parameter	Value	Wrapper/Payload
1	01	Format Version Number	1.00	Wrapper
2	00			
3	02	Message Flags	Ack Required	
4	07	Message Counter	2035	
5	F3			
6	00	Session Vehicle ID	127	
7	7F			
8	00	Optional Data Fields	0	
9	00			
10	0B	Message ID	11	Payload

Byte Number	Data (HEX)	Parameter	Value	Wrapper/Payload
11	00	Session Vehicle ID	127	
12	7F			
13	09	Message Timestamp	090616182030	Wrapper
14	06			
15	16			
16	18			
17	20			
18	30			

Log Off Response from Server

Byte Number	Data (HEX)	Parameter	Value	Wrapper/Payload
1	01	Format Version Number	1.00	Wrapper
2	00			
3	03	Acknowledge Flags	Ack	
4	F6	Acknowledgement Counter	63214	
5	EE			
6	07	Message Counter Ref	2035	
7	F3			
8	00	Session Vehicle ID	127	
9	7F			
10	09	Message Timestamp	090616182035	Wrapper
11	06			
12	16			
13	18			
14	20			
15	35			
16	0	Error Number	0 – No Error	

B.4 Basic Journey Details Message

Journey Details message from Bus

Byte Number	Data (HEX)	Parameter	Value	Wrapper/Payload
1	01	Format Version Number	1.00	Wrapper
2	00			
3	02	Message Flags	Ack Required	
4	04	Message Counter	1093	
5	45			
6	00	Session Vehicle ID	127	
7	7F			
8	00	Optional Data Fields	0	
9	00			
10	1F	Message ID	31	Payload
11	36	Service Code	631	
12	33			
13	31			
14	00			
15	00			
16	00			
17	43	Running Board	CA456	
18	41			
19	34			
20	35			
21	36			
22	00			
23	00			
24	34	Journey Number	42	
25	32			
26	00			
27	00			

Byte Number	Data (HEX)	Parameter	Value	Wrapper/Payload
28	00			
29	0B	Journey Scheduled Start Time	12:55	
30	37			
31	FF	Duty Number	Not Available	
32	FF			
33	FF			
34	36	Public Service Code	63A	
35	33			
36	41			
37	00			
38	00			
39	00			
40	01	Direction	1	
41	09	Message Timestamp	090616125030	Wrapper
42	06			
43	16			
44	12			
45	50			
46	30			

Acknowledgement from Server

Byte Number	Data (HEX)	Parameter	Value	Wrapper/Payload
1	01	Format Version Number	1.00	Wrapper
2	00			
3	03	Acknowledge Flags	Ack	
4	B7	Acknowledgement Counter	46976	
5	80			
6	04	Message Counter	1093	

Byte Number	Data (HEX)	Parameter	Value	Wrapper/Payload
		Ref		
7	45			
8	00	Session Vehicle ID	127	
9	7F			
10	09	Message Timestamp	090616125035	
11	06			
12	16			
13	12			
14	50			
15	35			
16	00	Error Number	0 – No Error	

B.5 Basic Position Update Message

Position Update Messages from Bus

Byte Number	Data (HEX)	Parameter	Value	Wrapper/Payload
1	01	Format Version Number	1.00	Wrapper
2	00			
3	02	Message Flags	Ack Required	
4	04	Message Counter	1061	
5	25			
6	00	Session Vehicle ID	127	
7	7F			
8	00	Optional Data Fields	0	
9	00			
10	29	Message ID	41	Payload
11	0B	Latitude	187200000	
12	28			

Byte Number	Data (HEX)	Parameter	Value	Wrapper/Payload
13	72			
14	00			
15	00	Longitude	7488000	
16	72			
17	42			
18	00			
19	5A	Bearing	180	
20	09	Message Timestamp	090616124100	Wrapper
21	06			
22	16			
23	12			
24	41			
25	00			

B.6 End Of Journey Message

Messages from Bus

Byte Number	Data (HEX)	Parameter	Value	Wrapper/Payload
1	01	Format Version Number	1.00	Wrapper
2	00			
3	02	Message Flags	Ack Required	
4	04	Message Counter	1211	
5	BB			
6	00	Session Vehicle ID	127	
7	7F			
8	00	Optional Data Fields	0	
9	00			
10	27	Message ID	39	Payload
11	36	Service Code	631	

Byte Number	Data (HEX)	Parameter	Value	Wrapper/Payload
12	33			
13	31			
14	00			
15	00			
16	00			
17	34	Journey Number	42	
18	32			
19	00			
20	00			
21	00			
22	0B	Journey Scheduled Start Time	12:55	
23	37			
24	36	Public Service Code	63A	
25	33			
26	41			
27	00			
28	00			
29	00			
30	01	Direction	1	
31	09	Message Timestamp	090616140530	Wrapper
32	06			
33	16			
34	14			
35	05			
36	30			

Acknowledgement from Server

Byte Number	Data (HEX)	Parameter	Value	Wrapper/Payload
1	01	Format Version	1.00	Wrapper

Byte Number	Data (HEX)	Parameter	Value	Wrapper/Payload
		Number		
2	00			
3	03	Acknowledge Flags	Ack	
4	BB	Acknowledgement Counter	48005	
5	85			
6	04	Message Counter Ref	1211	
7	BB			
8	00	Session Vehicle ID	127	
9	7F			
10	09	Message Timestamp	090616140535	
11	06			
12	16			
13	14			
14	05			
15	35			
16	00	Error Number	0 – No Error	