Application Communication Protocol
Application Layer Message Set Definition

<u>Version 3.1.0</u>

March 20, 2000

Prepared by
Motorola, Inc
4100 Commercial Ave
Northbrook, IL 60062

Revision History

Revision	Date	Author	Description
3.0.0.0	5/17/99	Tanya Bhaumik	Initial Version for the Interface Control Document for the Application Communication Protocol Application Layer Message Definition
3.0.0.2	5/21/99	Tanya Bhaumik	Update document based on ACP Working Group meeting on 5/18/99.
3.0.0.3	5/24/99	Tanya Bhaumik	Add Application Identifier(AID) for backward compatibility with the ACP transport layer.
3.0.0.x	7/2/99	John Emrich	Modification of the Provisioning section to include additional functionality for Fleet Management application.
		Samir Tailor	Modification and reorganization of document for maintainability and future additions application features.
3.0.0.4	8/16/99	Samir Tailor John Emrich	Modifications due to review, implementations, and feedback both internal and external
3.0.0.5	1/17/00	Mel Hagar	Added new and modified existing Fleet Management Application. Added Application Configuration Application. Incorporated editorial changes.
3.0.0.6	1/24/00	Mel Hagar	Incorporated comments from internal review.
3.1.0	3/20/00	Mel Hagar	Released 3.0.0.6.

Table Of Contents

1 Introduction

1.1 Purpose

It is the goal of the Application Communication Protocol Application Layer Document to establish the basis for agreement between the Telematics Communications Group and the service operator by providing a description of messages. This document also aims to document the protocol for data exchange between the TCU and service operator. This document will show what information will be made available to the service operator.

This document will describe the message exchange between the TCU and SO. Each message is described in detail and shall serve as the sole document for controlling the interface between the TCU and service operator. Changes to this interface or protocol shall be controlled and monitored by both the TCU provider and the service operator.

In this document this protocol is referred to as Application Communication Protocol or ACP.

1.2 Intended Audience

The Application Communication Protocol Application Layer Document is intended for:

- TCU vendor
- Service Operator
- Vehicle Manufacturer

1.3 Scope

This document describes the message set and message protocol for the ACP application layer. This document does not cover framing methods or checksums that deal with the transmission infrastructure. Those items are defined by the cellular infrastructure. This document assumes the existence of a reliable transport protocol underneath, such as defined in [[ACP TL]].

This document does not cover the details of the cellular infrastructure present within the TCU.

1.4 Change Process for this document

The intent is to make the ACP specification open and this specification is intended as the first version of open ACP V3. The maintenance and future of this protocol will be the responsibility of the open ACP forum and its participants. In the interim until the maintenance of this spec is conveyed to the ACP forum, changes and suggestions can be sent to a specified email address.

1.5 How to read this document

This document initially describes in section 5 the rules for defining information elements. It follows with a description of the application header in section 6 that is common to all application level messages. Then the applications and their messages are defined in sections 7 to 15. The information elements that are used to define the messages are defined and described in section 16.

The information elements can be optional in one of two ways. The first is indicated by [O]. The second is the length field does not include the element and all elements beyond the stated length, even though it is not marked with a [O]. The element may be of variable length as indicated by [V]. The absence of a [O] indicates the field is mandatory, and the absence of a [V] indicates the field has a fixed size. Messages consist of one or more information elements. An information element that is defined, but is not used in a message has a length field of zero. The kind of information element is denoted by its position in the message and therefore, unused elements still must be included.

2 References

2.1 Normative References

[ACP TL]	Application Communication Protocol Transport Layer

2.2 Informative References

This document describes the data format and message protocol on the application layers. The

This document does not cover framing methods or checksums that deal with the transmission infrastructure. Those items are defined by the cellular infrastructure.

This document does not cover the details of the cellular infrastructure present within the TCU.

[RFC908]	"Reliable Data Protocol", RFC908, 07/01/1984, R. Hinden, J. Sax, D. Velten
[RFC1151]	"Version 2 of the Reliable Data Protocol (RDP)", RFC1151, 04/05/1990, R. Hinden, C. Partridge
[RFC2119]	“Key words for use in RFCs to Indicate Requirement Levels”, RFC2119, March 1997, S. Bradner
[ETS 300 642]	Digital Cellular Telecommunication System(Phase 2); AT command set for GSM Mobile Equipment(ME)
[Ullman]	“Principles of Database and Knowledge – Base Systems”, Volume 1: Classical Database Systems, 1995, Jeffry Ullman
[UNICODE]	International Standard ISO/IEC 10646-1:1993
	Applicable System Specifications

3 Definitions and abbreviations

All non-trivial abbreviations and definitions used in this document are listed in the following sections. The definition section includes description of general concepts and issues that may be fully defined in other documents. The purpose of this section is merely to advise the reader on the terminology used in the document.

3.1 Definitions

The notation used in the specification part of this document uses the common elements defined here. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Term	Definition
Entity	An abstract addressable process or node on a network. In OSI terminology an entity is an active network element.
Heading	Horizontal direction expressed as the angular distance in degrees between the direction of a fixed point (true North) and the direction of the object. North has an heading of 0°, East is 90°, South is 180° and West is 270°.
Protocol	Rules used to negotiate the exchange of specific type of information
Public Emergency Service Center	The initial point of contact when a user makes a voice call to the emergency number (112 in Europe, 911 in North America). This term is more frequently used in Europe for PSAP.
Public Service Access Point	The initial point of contact when a user makes a voice call to the emergency number (112 in Europe, 911 in North America).
Service Operator	A private Service Access Point for message exchange.
Telematics	the delivery and collection of location dependent services
Telematics Control Unit	Telematics unit is a device using a wireless communications using telematics.
Terminal	A device typically used by a user to request and receiving information. Also called a mobile terminal or mobile station.
User	A user is a person who interacts with a user agent to view, hear or otherwise use a resource.
Virtual Circuit	Type of service available for packet switch data networks and circuit switch data networks that first establishes a communication path through the network and then exchange messages between two connecting points.

3.2 Abbreviations

The following abbreviations apply to this document.

Abbreviation	Definition
ASN.1	Abstract Syntax Notation 1
ACP	Application Communications Protocol
AMPS	Advanced Mobile Phone System
BER	Bit Error Rate
BS	Bearer Services (refers to both BS24 & BS26)
BS24	Bearer Services 2400 bps

Abbreviation	Definition
BS26	Bearer Services 9600 bps
CLIP	Calling Line Identification Presentation (like Caller ID in the US)
CRC	Cyclic Redundancy Check
DOP	Dilution of Precision
ECC	Error Checking and Correction
ECN	Emergency Call Number
FEC	Forward Error Correction
FEC-15	short hand for RS(255,225,15) encoding
FEC-2	short hand for RS(255,252,2) encoding
GMT	Greenwich Mean Time
IE	Information Element
IMEI	International Mobile Equipment Identification
IMSI	International Mobile Subscriber Identification
LSB	Least Significant Bit
MMI	Man Machine Interface
MO	Mobile Originated
MSB	Most Significant Bit
MT	Mobile Terminated
NAM	Number Assignment Module
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
PSAP	Public Service Access Point
RFC	Request For Comments
RS	Reed-Solomon FEC encoding/decoding
RF	Radio Frequency
SMS	Short Message Service (both Mobile Originated and Mobile Terminated)
SMSC	Short Message Service Center
SO	Service Operator
TCU	Telematics Control Unit
TE	Terminal Equipment
UTC	Coordinate Universal Time

4 Requirements

These requirements are deduced from the environment in which ACP will execute. This defined protocol must work correctly at both the mobile unit and at the service operator. For example the service operator must be able to concurrently receive multiple messages from different mobile units. A large message can be transmitted as multiple packets. Optionally a set of consecutive small messages can be transmitted in a single transmission unit. The protocol must support this capability. The protocol will provide the capability to switch between voice and data calls.

The applicable system specifications must be referenced to determine which applications are being used and the method of use. There are optional messages and optional elements within each message. The system specification should call out which optional messages are being used and their usage scenarios.

4.1 Application Level

Table 1: details, but does not limit, the telematics applications that are available to the vehicle.

Application ID	Definition	Brief description
1	Provisioning (Section 7)	A basic application whereby the TCU establishes and manages configurations for applications required for activation. An example is updating the phone numbers for contacting the Service operator.
2	Configuring (Section 16)	This is provided as additional means of managing applications that are not handled by the generic provisioning applications, but are handled by the application.
3	Emergency Call (Section 8)	This application requires response from either police, fire brigade, or medical support.
4	Roadside Assistance Call (Section 9)	This application requires use of services such as tow truck or services of a mechanic due to vehicle problems.
5	Information Call (Section 10)	This application requires no on site assistance by the caller. The caller only requires some information that can be conveyed in a voice call. An example is tourist info. This is sometimes referred to as concierge service.
6	Remote Vehicle Function (Section 11)	This application allows control functions to occur within the vehicle. Examples are remote door unlock.
7	Fleet Management (Section 15)	This application can gather data on the use of its fleet of vehicles as well as track individual vehicles.
8	On-Board Navigation	This application provides the capability to navigate the vehicle using on-board equipment.
9	Off-Board Navigation	This application provides the capability to provide navigation information from a remote location.
10	Vehicle Tracking (Section 13)	This application provides the ability to track a vehicle, usually due to theft.
11	Alarm Indication (Section 12)	This application provides the ability to recognize unauthorized starting or movement of a vehicle.

Application ID	Definition	Brief description
12	Tele-Diagnostics (Section 14)	Provides for 1. TCU diagnostics 2. Vehicle diagnostics 3. Black box
13	Reserved	
14-63	Reserved; not yet assigned	

Table 1: Application Definitions

Each of the listed applications must be able to operate without any of the other telematics applications. These applications are defined so as to allow concurrent operation. For instance if an emergency call becomes active, it takes precedence over roadside assistance. But information call can run concurrent with fleet management

A telematics communications unit, TCU, is not limited to being in a car. Some of the other vehicle types in which it can execute are: bus, truck, and taxi. The TCU requires some kind of wireless device in order to accomplish its function. It is conceivable, that a limited capability TCU can be combined with a wireless device, be transportable, not be in a vehicle, but be carried about with the user.

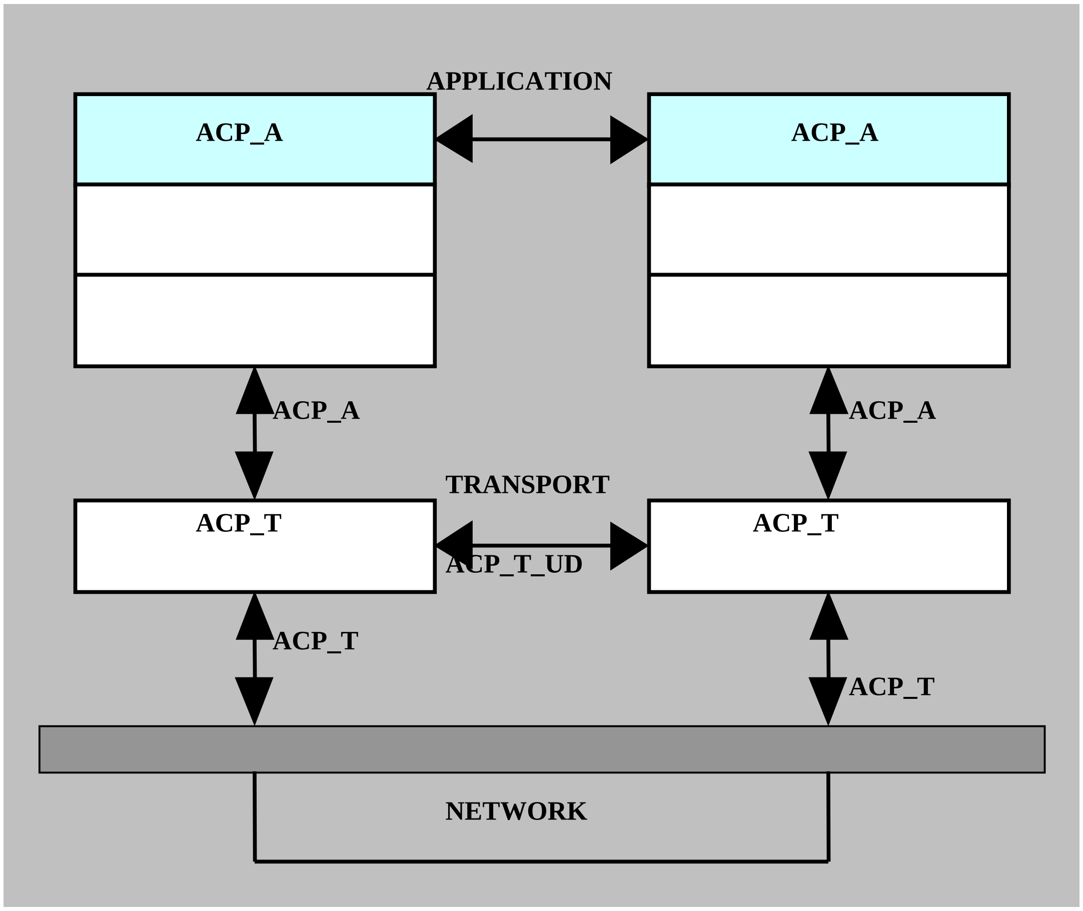
4.2 Transport level (Informative)

The ACP Transport Layer Document [ACP TL] defines the behavior of the transport layer. The entire application data contained in the application message is communicated to and from the transport layer. The high-level application layer will present a T_DATA.req to the transport layer. The remote application layer will receive the message as a T_DATA.ind message from the local ACP transport layer. The remote application layer will respond to the message it received by making a T_DATA.res request to its local transport layer. The transport layer is then responsible for sending the data to the originator. The originator upon receiving the data at the transport layer will forward the data as a T_DATA.cnf to the originating application layer.

4.3 Model for ACP

This section is from reference [[ACP TL]] and illustrates the interrelationships.

The model for the protocol makes use of service primitives. The definitions beginning with the prefix ACP_A_ are used to exchange information between the ACP and the applications. They are the means through which the ACP provides services to the application and the application invokes the services of the ACP. The definitions beginning with ACP_ are used for peer to peer exchange of packets and control information between ACP instances.



5 Data Description Rules (Normative)

5.1 Rules

This section describes the message formats used in the exchange of data between a TCU and the SO. Message formats consist of information elements as needed to create a meaningful message. This standard defines the minimal set of messages necessary to support the telematics functions. Undefined and non-supported messages and information elements can be skipped. The intent is to allow for the addition of messages information elements without affecting existing applications of current customers. This rule is intended to promote robustness and is not intended to encourage sending useless information between TCU and SO.

The basic unit of data is an information element. Information elements can be combined to form message types, the unit used for the transfer of data by the transport level. A number of message types are defined in this standard along with their use.

We use the Basic Encoding Rules, of ASN.1 to describe the different message sets and information groups. The definitions are of the form TLV where T designates type, L designates length and V designates the value.

The standard information elements are defined:

- T: This format defines an optional message set or information group. The T defines the message type.
- TL: This is used to define an optional variable length message set or information group where the length is variable. The T defines the message type. The L defines the length of the message set.
- V: This defines the value for the information element.

The length field, L, when used can range from 0 (no data present) to the maximum length of the field. The length is always in octets unless specifically stated otherwise. An optional message set is defined to have length 0 when it is not present.

5.2 Bit Order

Bits in a octet are numbered from 0 to 7 from most significant bit to least significant bit. Multi-octet fields have the most significant octet in the in the first octet of the field.

5.3 More Flag

If the value is 0, then this is the last octet of continuous information in the header before the next field. If the value is 1, then another information octet follows this octet. In the following example, the length is 12 bits with the most significant bit defined by bit 3 in the first octet. A more flag of zero is in the last length octet indicating that this octet is the last octet of length.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE identifier		More Flag=1	Length (number of octets that follow)				
2	More Flag	Length						

5.4 Element Definition

The information elements are identified by a 2 bit *IE identifier, More flag* and a *length*, followed by the data. The *length* field specifies the length of the following field. If the field size exceeds 32 octets, then the *More flag* is set. This indicates the following element specifies a continuation of the current element. There is no limit on the number of continuation elements, just a practical limit.

An empty information element is one octet with a length field of zero. If the IE identifier field contains a 3, then the extended definition is used and an empty information is 2 bytes long with the length byte of 0 again. An application can skip an information element by just looking at the length field.

IE and Length information is present when specified in the ACP message or in the element definition itself. Otherwise the IE and Length information is not included. For example, the Absolute Time element does not have an IE and length field.

<table>
<tr><th>Octet \ bit</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th></tr>
<tr><td>1</td><td colspan="2">IE identifier</td><td>More Flag</td><td colspan="5">Length (number of octets that follow)</td></tr>
<tr><td>2..n-1</td><td colspan="8">Encoded String</td></tr>
</table>

Table 2: Octet string definition

The current values for IE Identifier are:

IE Identifier	Type	Meaning of length field
Transparent data (binary data)	0	Number of 8 bit octets
Text, 8 bits per character (ISO 8859-1)	1	Number of 8 bit octets
packed decimal	2	Number of 8 bit octets
Extended string definition	3	Extended string definition

Table 3: IE Identifier

The following table defines an extended information element for string.

<table>
<tr><th>Octet \ bit</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th></tr>
<tr><td>1</td><td colspan="2">3 (Extended string definition)</td><td colspan="6">IE identifier</td></tr>
<tr><td>2</td><td>More Flag</td><td colspan="7">Extended IE identifier</td></tr>
<tr><td>3</td><td>More Flag</td><td colspan="7">Length (number of octets that follow)</td></tr>
<tr><td>4..n-1</td><td colspan="8">Encoded String</td></tr>
</table>

Table 4: Extended IE Identifier Definition

The current values for Extended IE identifier are:

IE Identifier	Type	Meaning of length field
Transparent data (binary data)	0	Number of 8 bit octets
Text, 8 bits per character (ISO 8859-1)	1	Number of 8 bit octets
packed decimal	2	Number of 8 bit octets
Reserved	3	
UNICODE ver 2 text data (ref UNICODE)	4	Number of 8 bit octets
Utf-8	5	Number of 8 bit octets
Shift JIS	6	Number of 8 bit octets
Reserved	7 to 61	TBD
Extended private identifier	62	TBD
Extended public identifier	63	TBD

Table 5: Extended IE Identifiers

5.5 IE Identifier

The values of 62 and 63 have the following meaning. If the value is 62, then this element is not defined in this document and is considered proprietary. The private identifier is in the following octet. If the value is 63, then an octet of extended identifiers is in the following octet.

6 Application Header

Each message between the TCU and SO contains an Application Header defined in the tables below. The header used is dependant on the underlying transport layer used and version of the header. Tables 6 and 7 are used when the ACP application protocol is used with the ACP transport protocol. Tables 8 and 9 are used with other reliable transport protocols such as TCP/IP

Octet \ bit	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	Reserved Set to 0	Non Standard Flag	Test Flag	Message Type (1-31)				
2	Version Flag	Version (6.2)			Message Control Flag (6.4)			
3-4	Message Length (6.6)							

Table 6: Application Header for ACP Transport(Version Flag = 0)

Octet \ bit	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	Reserved Set to 0	Non Standard Flag	Test Flag	Message Type (1-31)				
2	Version Flag	Version (6.2)			Message Control Flag (6.4)			
3 (optional)	More Flag	Reserved Set to 0					Message Priority Flag	
4-5	Message Length (6.6)							

Table 7: Application Header for ACP Transport Header (Version Flag=1)

Octet \ bit	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	Reserved Set to 0	Non Standard Flag	Application ID (1-63) (4.1)					
2	Reserved Set to 0	Non Standard Flag	Test Flag	Message Type (1-31)				
3	Version Flag	Version (6.2)			Message Control Flag (6.4)			
4-5	Message Length (6.6)							

Table 8: Application Header for Other Transport Protocols (Version Flag = 0)

Octet \ bit	0	1	2	3	4	5	6	7
1	Reserved Set to 0	Non Standard Flag	Application ID (1-63) (4.1)					
2	Reserved Set to 0	Non Standard Flag	Test Flag	Message Type (1-31)				
3	Version Flag	Version (6.2)			Message Control Flag (6.4)			
4 (optional)	More Flag	Reserved Set to 0					Message Priority Flag	
5-6	Message Length (6.6)							

Table 9: Application Header for Other Transport Protocols (Version Flag=1)

6.1 Message Type

This field defines the message type. It is unique for a specific message. If a message is constructed with a unique combination of message elements then that message will have a unique message type. The definition of the message is defined by the application. If the *Non Standard Flag* is set, then this message type is a custom message. If the *Test Flag* bit is set, then this is an application test message.

6.2 Version (Version Flag=0)

This is the version of the application as defined by the application. This allows the application to be revised and identified independent of the overall ACP version number. This field is used when subsets or super sets of the same application are defined. This field is initially set to the same version as the ACP version 3.

6.3 Version Flag

This is intended to allow for extended control information in the header that is common among many applications. If the value is 0, then this is the last byte of the header prior to the field *message length*. If the value is 1, then another control octet of information follows this octet.

6.3.1 ACP Version 2 TCU Migration Issues

It is critical to maintain backward compatibility when migrating from ACP Version 1 to ACP Version 2 to ACP Version 3. The format of ACP application message set for Version 1 and Version 2 are defined in section 19 for emergency call and section 18 for authorization application. To do this requires an interface to the transport layer (ref [ACP TL]) whereby the transport layer informs the application if the incoming message is version 1, 2, or 3.

6.4 Message Control Flag

These bit definitions define various actions that are common over many applications.

Bit number	Definition
0	Reserved, set to 0
1	0=Use TLV as defined in section 5
2	1=The Message Length field is 16 bits 0=The Message Length field is 8 bits
3	1=Application level response is expected 0=Application level response is not expected

Table 10: Message Control Flag (VersionFlag=0)

Bit 3 indicates the sender is expecting an application level response message from the receiver. In an example from provisioning, this is set by the SO when sending a *provision update message #1*, because the SO is expecting the TCU to send a *provision reply message*. Again the TCU sets the bit when sending its *provision reply message* when the TCU is expecting the optional *provision commit message*.

6.5 Message Priority Flag

This field is optional and is used to define certain actions

Name	Value	Definition
	0	Not currently used
Abort	1	This application is being aborted by the sender of this message. The receiver should also abort its application. The field *message length* may be 0. If it is not 0, then the field *error element* follows (17.13)
Pause	2	This application is being paused because another application with higher precedence must run
Resume	3	An application that was paused before can start up again.

Table 11: Message Control Flag (Version Flag=1)

6.6 Message Length

This field can have values from 0 to 255 octets. If this field is defined as 16 bits, then the range is from 0 to 65,535. The message length is the total number of octets in the message including the header.

6.7 Bearers used

For some applications it may be desirable to support multiple bearers in a single invocation. This requires an API to the transport layer to provide the application layer with information as needed in order to control this.

Receiving messages over multiple bearers can be an issue. An accurate caller ID service can provide this in the call setup request for CSD type bearers. This number is present for most bearers in most situations. However, there are cases such as call forwarding when caller ID is not reliable. It is left as an implementation decision to decide whether an incoming call on a CSD bearer is intended for telematics or a different application.[1]

[1] For example in GSM using SMS messages, class 3 binary plus the unique PID is usually adequate to

6.8 Non-Standard Flag

If the value is 0, then this application message is defined in this document. If the value is 1, then this application message is not defined in this document and is considered proprietary.

6.9 Test Flag

If the value is 0, then this application message is to be acted upon. If the value is 1, then this application message is a test message and action is dependent upon the implementation.

distinguish an incoming message for ACP instead of a non APC application.

7 Provisioning Application

7.1 Provision protocol (Normative)

These are messages that allow the service operator (SO) and the TCU in the vehicle to provision applications. This allows for the establishment of features and services by application . Parameter updates are usually done within an application. This can include subscription information depending on the business model of the SO. It can cover the means whereby the TCU customer pays (e.g. on a usage basis or for a fixed period basis or some other combination). If the service operator wants to sign up a customer when the service is 1st used, then more information must be supplied as part of the application, and information the customer may not readily have at time of use.

We are only concerned with defining the message exchange necessary to support basic accounting/usage models. The actual business models are the purvey of the service operators.

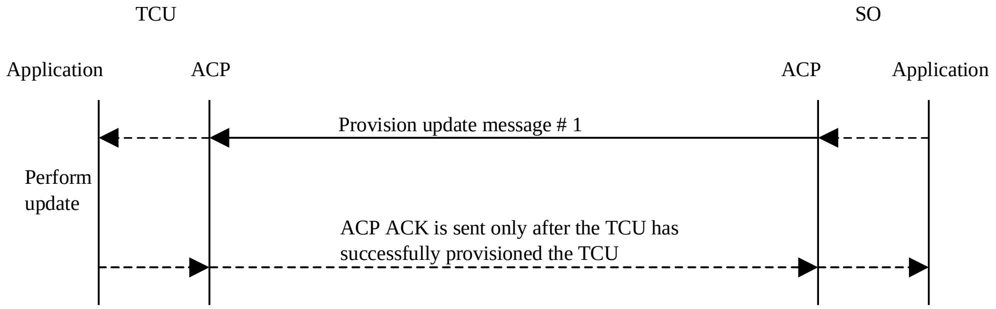
A TCU must initially be provisioned with sufficient information to either accept a *Provision update message* or send a *Provision update request message*. It is up to the vendor to pick the approach used. The *Provision update message* allows the SO to begin the update sequence whereas the *Provision update request message* allows the TCU to begin the update sequence. Since the SO will not usually know when the TCU is available for *Provision update message*, provision updates are usually initiated by the TCU. However, for those bearers, such as SMS where the message can be stored in the wireless network, the SO may choose to initiate provision update with a *Provision update message*. The approach used will be dictated by the wireless network bearer used.

An optional capability exists that allows the TCU to initiate the provisioning exchange. The TCU must be initialized with sufficient information to contact the SO for the service at the time the service is needed. Without this initial information, the TCU does not know how to contact the SO. This message exchange is normally preceded by some business agreement between TCU owner and SO operator.

This protocol can be used to update the cargo description of the vehicle as needed for an application

7.2 Example of very simple SO initiated Provisioning

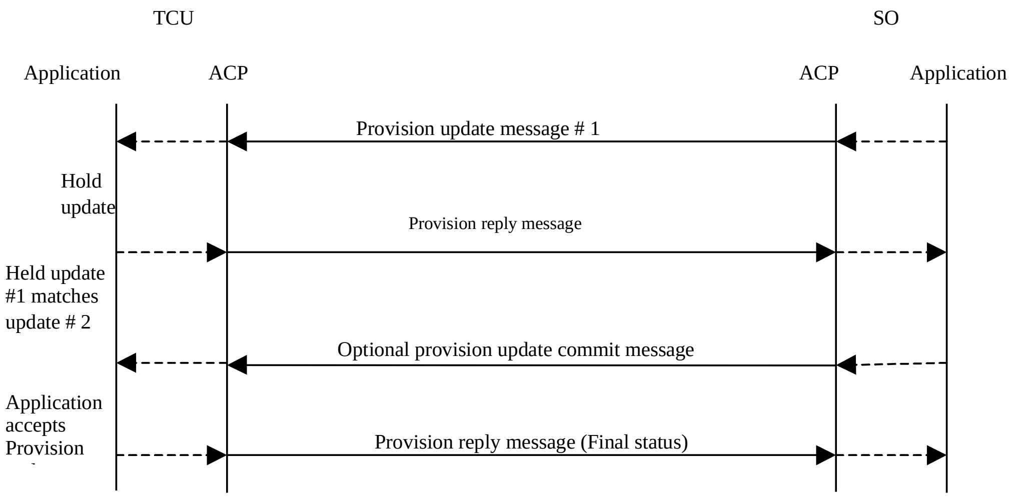
This example assumes a reliable media for provisioning the TCU. The provisioning application in the TCU does not have a separate confirmation message, but uses the transport acknowledgement to inform the SO of successful provisioning.


TCU
SO
Application
ACP
ACP
Application
Provision update message # 1
Perform update
ACP ACK is sent only after the TCU has successfully provisioned the TCU

7.3 Example of SO initiated provision message exchange

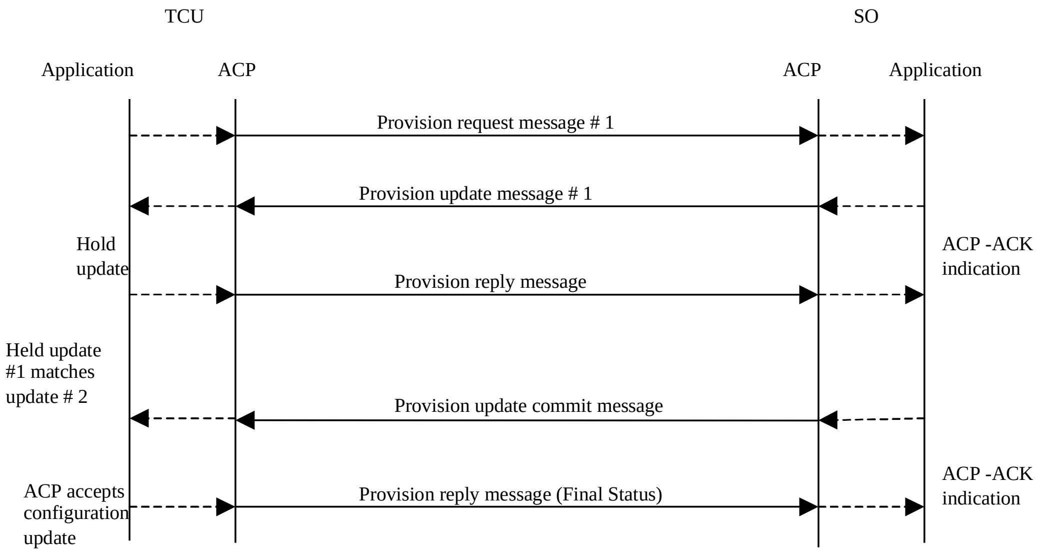
This is an example of the initial message exchange that is necessary for the SO to provision the TCU with the appropriate information [e.g. phone numbers] so that it may place an emergency call.

The *provision update commit message* is available for those applications that want a more robust provisioning mechanism. It is based on distributed data base technology using atomic 2-phase commit (see ref [Ullman]). This is especially important for SMS when realizing that when the SO sends a message to the TCU, the TCU may be turned off and the SMS message is queued at the SMSC. When *provision update commit message* is used, then the sender must receive some acknowledgement that the *provision update commit message* was successful. This provides for a reliable provisioning of the TCU if required..


TCU
SO
Application
ACP
ACP
Application
Provision update message # 1
Hold update
Provision reply message
Held update #1 matches update # 2
Optional provision update commit message
Application accepts Provision
Provision reply message (Final status)

7.4 Example of TCU initiated provision message exchange

This is an example of a TCU requesting an update of provision from the SO. An example of this is the TCU’s service agreement is expiring and the TCU’s owner wants to renew the agreement. The SO may choose to renew or not renew.


TCU
SO
Application
ACP
ACP
Application
Provision request message # 1
Provision update message # 1
Hold update
ACP -ACK indication
Provision reply message
Held update #1 matches update # 2
Provision update commit message
ACP accepts configuration update
Provision reply message (Final Status)
ACP -ACK indication

7.5 Provision message set (Normative)

These are the actual messages required to accomplish a provisioning call.

Message Type	Description
1	Provision Update Message #1 (7.5.1)
2	Optional Provision Update Commit Message (7.5.2)
3	Provision Reply Message #1 (7.5.3)
4	Optional Provision Reply Commit Message #2 (7.5.3)
5	Provision Request Message (7.5.4)
6	Provision Status Message (7.5.4)

7.5.1 Provision update message #1

This message is sent from SO to TCU to update information in the TCU. It can be used to update the vehicle parameters.

Octet \ bit	0	1	2	3	4	5	6	7
1.3	Application header (6)							
4…8	Version (17.4) [OV]							
9	More Flag	Target Application ID (7.5.1.1)						
10	ApplFlag1 (17.1)		ControlFlag1(17.2)					
11	ControlFlag2 (17.3) [O]							
12	Start Time (17.5) [O]							
16	End Time (17.6) [O]							
20	Grace Time (17.6) [O]							
24	TCU Descriptor Element (17.8) [OV]							
25-n-1	Vehicle Descriptor (17.23) [OV]							
n..m-1	Phone Number (17.34) [OV]							
:	:							
j..k-1	Phone Number (17.34) [OV]							

Table 12: Provision update message

7.5.1.1 Target Application ID

This field takes on the same values as application ID (4.1). It is used to identify the application that is to be provisioned. The More Flag is used to provide a list of Application Ids to be provisioned with the same data.

The More Flag is used to provide a list of Application Ids to be provisioned with the same data. The Provisioning and Configuring target Applications Ids should not be used in the Provision update message.

7.5.1.2 Example 1:

For simple application, the specified numbers are for SMSC bearer, SMS bearer, Voice, then 112 voice.

7.5.1.3 Example 2:

For an advanced application that provides for a backup SMS bearer, the specified numbers are for primary SMSC bearer, primary SMS bearer, secondary SMSC bearer, secondary SMS bearer, primary voice, secondary voice, then 112 voice.

7.5.2 Provision update commit message

The format of this message is exactly the same as *provision update message #1*. However, the value of *ApplFlag1* can differ based on the *provision reply message* received by the SO. Thus when a system uses this message they are deferring the provisioning until the SO knows the TCU can perform the requested *provision* action. This is most important when using SMS and using long time to live for SMS messages.

7.5.3 Provision reply message

The message is similar to the *Provision update* message except the *Phone Number* elements are not present. The *Expiration Time* is updated by the TCU. This detail allows the SO to verify the data is the same as that which was sent. It provides for additional robustness in case messages are duplicated.

Octet \ bit	0	1	2	3	4	5	6	7
1-3	Application header (6)							
4…8	Version (17.4) [OV]							
9	More Flag	Target Application ID (4.1)						
10	ApplFlag1 (17.1)		ControlFlag1(17.2)					
11	StatusFlag1 (17.10)		TCU Response Flag (17.11)		Reserved, set to 0			
12	Start Time (17.5) [O]							
16	End Time (17.6) [O]							
20	Error Element (17.13) [O]							
21	Vehicle Descriptor Element (17.8) [OV]							

Table 13: Provision reply message

The fields *Target Application ID, ApplFlag1*are exactly what was sent in the *Provision update message #1*. If no *provision update commit message* is expected, then a value of *OK* in the *error element* indicates the result of the provisioning request. Sending a *Provision update message* with *ApplFlag1* set to *no change* allows the SO to query about the provisioning status of a TCU.

7.5.4 Provision request message and provision status message

Both messages have the same format. This message is sent from the TCU to the SO to request a provision update. It is typically used if a unit has been authorized for a service, and is either nearing end of that authorized period or has elapsed into a grace period. The format of the response is identical to *Provision reply* message except the message id indicates a *Provision request* message.

The *provision status* message is used at the end of a provisioning sequence so that the SO can verify the correct status.

If the provision request message is sent from SO to TCU, the SO is requesting a *provision status* message. This allows the SO to determine the provisioning status of an application in a TCU.

Octet \ bit	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1-3	Application Header (6)							
4…8	Version (17.4) [OV]							
9	More Flag	Target Application ID (4.1)						
10	ApplFlag1 (17.1)		ControlFlag1(17.2)					
11	StatusFlag1 (17.10)		Reserved, set to 0					
12	Start Time (17.5) [O]							
16	Expiration Time (17.11) [O]							
20	Error element (17.13) [O]							
21	Vehicle Descriptor Element (17.8) [OV]							

Table 14: Provision status request

8 Emergency Call Application

8.1 Emergency Call Message Protocol (Informative)

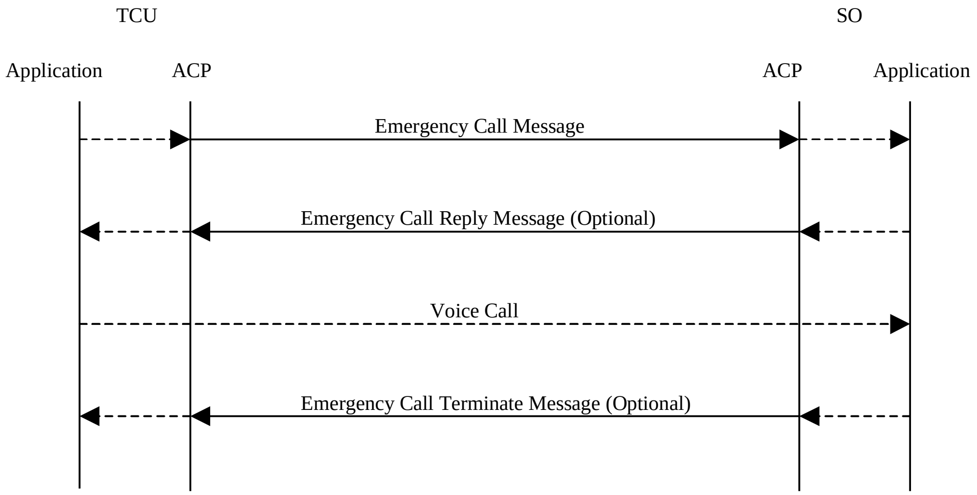
The emergency call application allows the TCU to request emergency assistance from the SO. Typical assistance required may include police, fire or medical assistance. The emergency call may be triggered manually via a pushbutton in the car or automatically (e.g. sensors or triggers).

In general, a request is made to the SO by the TCU for an emergency assistance. The SO then collects the required information and dispatches the appropriate services.

8.2 Example Emergency Call Message Exchange

The following is an example of an emergency call application message exchange. An emergency call message, triggered by a vehicle event, is sent from the TCU to the SO. The SO then sends an application layer reply back to the TCU to confirm message receipt. The SO then dispatches the appropriate services. The TCU then initiates a voice call with the SO using the number sent in the application reply message. See the figure below for an illustration of this message exchange.

When the TCU requires a reply message, it should set the response expected flag (6.4) to get the Emergency Call Reply Message with the phone numbers. Otherwise, the voice call can be initiated immediately.


TCU
SO
Application
ACP
ACP
Application
Emergency Call Message
Emergency Call Reply Message (Optional)
Voice Call
Emergency Call Terminate Message (Optional)

8.3 Emergency Call Message Set (Normative)

These are the actual messages required to accomplish an emergency call.

Message Type	Description
1	Emergency Call Request (8.3.1)
2	Emergency Call Reply (8.3.2)
4	Emergency Call Terminate (8.3.3)

Table 15: Emergency Message Set

8.3.1 Emergency Call Request Message

This message is a request from the TCU to the SO for emergency assistance.

Octet \ bit	0	1	2	3	4	5	6	7
1…3	Application Header (6)							
4…8	Version (17.4) [OV]							
9…12	Time Stamp (17.29)							
13…n-1	Location (17.14) [V]							
n…m-1	Vehicle Descriptor (17.23) [V]							
m…k-1	Breakdown Status (17.24) [V]							
k…i-1	Information Type (17.38) [O]							

Table 16: Emergency Call Request Message

At a minimum, the TCU must provide the SO with the date, time, location, vehicle description, and breakdown status. The breakdown status may give insight as to nature of the call and help response units to better equip themselves. The vehicle may or may not have the ability to support the optional fields. In the event the vehicle does support the detection of the optional fields, the TCU shall report them. The vehicle shall set the length of the optional fields to 0 (zero) otherwise.

The information type field is intended for use with the Information Call Application (10). This field specifies the type information that is being requested.

8.3.2 Emergency Call Reply Message

This message is a reply from the SO to the TCU confirming receipt of the emergency assistance request. This can also be used to re-request the current location by setting the appropriate bits in ControlFlag2.

Octet \ bit	0	1	2	3	4	5	6	7
1…3	Application Header (6)							
4…8	Version (17.4) [OV]							
9	Confirmation (17.31)				Transmit Units (17.33)			
10	ControlFlag2 (17.32)							
10..j	Error Element (17.13) [O]							
j…n-1	More Flag	Reserved	Transmit Interval					
n	Phone Number (17.34) [O]							
n+1…m-1	More Flag	Reserved	Transmit Interval					
m...k-1	Nth Phone Number (17.34) [O]							

Table 17: Emergency Call Reply Message

This message is always sent by the SO in response to the emergency call request. It is used to inform the TCU whether or not the emergency request can be processed. If the emergency notification can not be processed, then it specifies information whereby the TCU can proceed to its next course of action.

Some of the possible reasons for rejection are:

- The vehicle is in a country different from the SO
- The vehicle is not authorized for the service at this SO

- Use alternate SO

8.3.3 Emergency Call Terminate Message

This message is used by the SO to inform the TCU that the SO has terminated the voice call and the TCU should resume prior activities.

Octets \ bit	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1...3	Application Header							
4..8	Version element (17.4) [O]							
9	IE Identifier		More Flag	Length				
10..n	Car Manufacturer ID (17.4.3) [OV]							
n..m-1	TCU manufacturer ID (17.4.4) [OV]							
m..j-1	Major hardware release (17.4.5) [OV]							
j..k-1	Major software release (17.4.6) [OV]							

9 Roadside Assistance Call Application

9.1 Roadside Assistance Call Message Protocol (Informative)

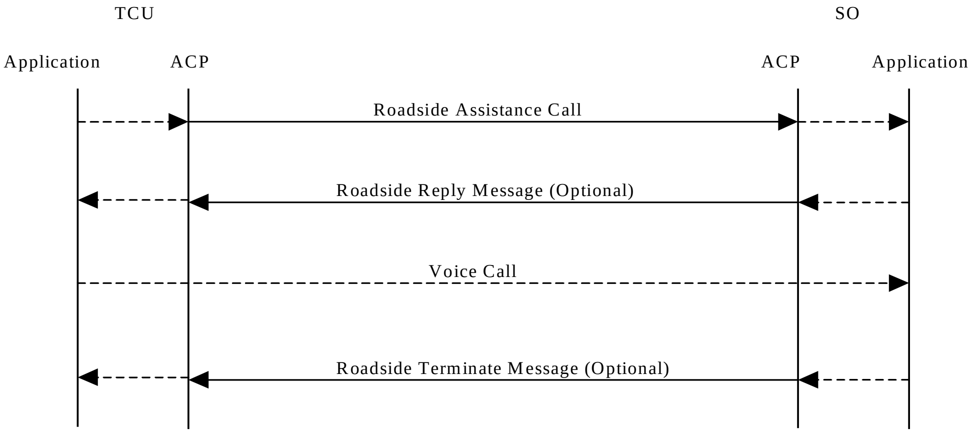
The roadside assistance call application allows the end user or the vehicle to request non-emergency roadside assistance from the SO. Typical assistance required may include a tow truck or a mechanic. The roadside assistance call is triggered manually via a pushbutton in the car.

In general, a request is made to the SO by the TCU for a roadside assistance. The SO then collects the required information and dispatches the appropriate services.

9.2 Example Roadside Assistance Call Message Exchange

The following is an example of a roadside assistance call message exchange. A roadside assistance call message, triggered by the end user, is sent from the TCU to the SO. The SO then sends an acknowledgement back to the TCU to confirm message receipt. The SO then dispatches the appropriate services. The TCU then initiates a voice call with the SO. Below is the message sequence illustrating this message exchange.

When the TCU requires a reply message, it should set the response expected flag (6.4) to get the Roadside Call Reply Message with the phone numbers. Otherwise, the voice call can be initiated immediately.


TCU
SO
Application
ACP
ACP
Application
Roadside Assistance Call
Roadside Reply Message (Optional)
Voice Call
Roadside Terminate Message (Optional)

9.3 Roadside Assistance Call Message Set (Normative)

These are the actual messages required to accomplish a roadside assistance call. The roadside assistance message set is the same as the emergency call message set (8.3). The distinction is made in the Application ID field of the ACP Transport Header[ACP TL]. Additionally, the information type field is not used in the roadside assistance application and thus this is not included in the message by setting the field length to 0.

Message Type	Description
1	Roadside Assistance Call Request (8.3.1)
2	Roadside Assistance Call Reply (8.3.2)
3	Roadside Terminate (8.3.3)

Table 18: Roadside Assistance Message Set

10 Information Call Application

10.1 Information Call Message Protocol (Informative)

The information call application allows the end user or the vehicle to request information from the SO. Typical information required may include nearest hotels, restaurants, or driving directions. The information call is triggered manually via a push button in the car.

In general, a request is made to the SO by the TCU for information assistance. The SO then collects the required information and responds verbally with the appropriate information.

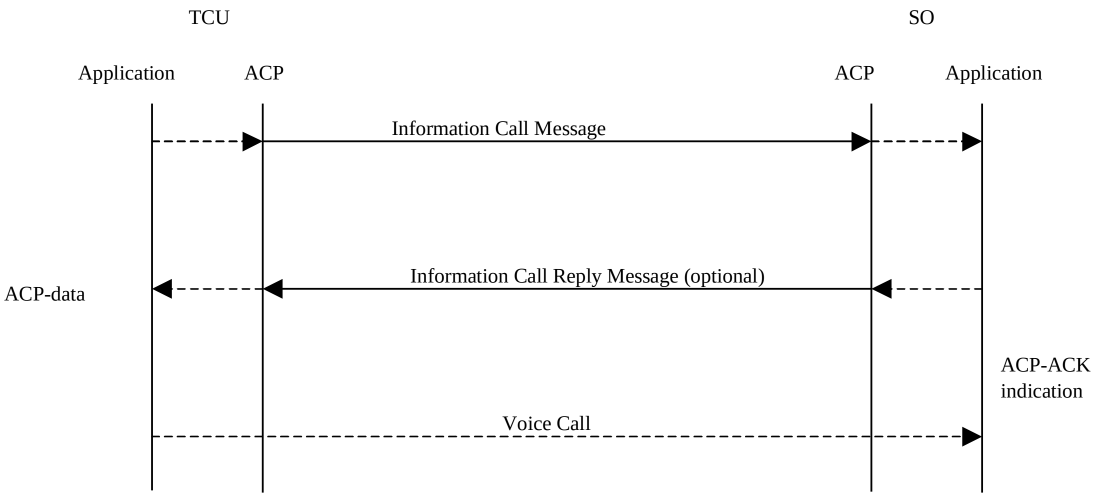
10.2 Example Information Call Message Exchanges

It is possible that the information requested can only be provided by the SO via a voice call. This is the case in most instances. However, it is also possible that the information requested can be sent to the TCU via a message. The following sections illustrate the two examples of message exchanges between the SO and TCU.

10.2.1 Example of Information Call Via Voice Call

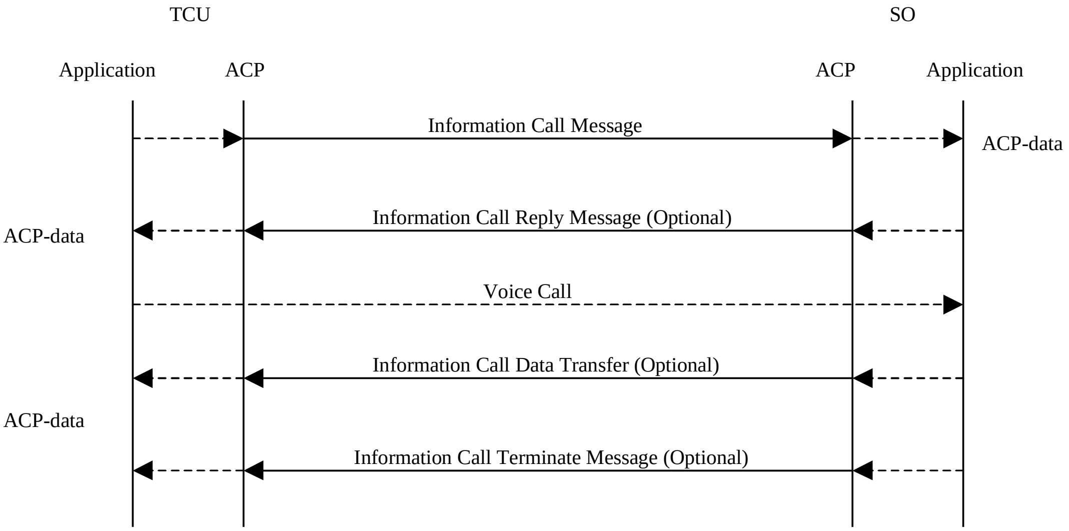
The following is an example of an information call message exchange accomplished via a voice call. An information call message, triggered by the end user, is sent from the TCU to the SO. The SO then sends an acknowledgement back to the TCU to confirm message receipt. The TCU then initiates a voice call with the SO. The SO then responds with the appropriate information. Below is the message sequence illustrating this message exchange.

When the TCU requires a reply message, it should set the response expected flag (6.4) to get the Information Call Reply Message with the phone numbers. Otherwise, the voice call can be initiated immediately.



10.2.2 Example of Information Call Via Data Call

The following is an example of an information call message exchange accomplished via a data call. An information call message, triggered by the end user, is sent from the TCU to the SO. The SO then sends an acknowledgement back to the TCU to confirm message receipt. The TCU then initiates a voice call with the SO. The SO determines what information is required and responds with a information call data transfer message to the TCU containing the information requested. Below is the message sequence illustrating this message exchange.


TCU
SO
Application
ACP
ACP
Application
Information Call Message
ACP-data
ACP-data
Information Call Reply Message (Optional)
Voice Call
Information Call Data Transfer (Optional)
ACP-data
Information Call Terminate Message (Optional)

10.3 Information Call Message Set (Normative)

These are the actual messages required to accomplish an information call. Depending on the information being requested, there may be multiple reply messages that are sent back to the TCU. The information call message set is the same as the emergency call message set (8.3) with an additional data transfer message. This message transfers the data that was originally requested by the TCU. Additionally, the information type field is required for this application. It is possible that the end user may wish to request information on a per use basis rather than a fixed rate. This can be accomplished by including the optional vehicle descriptor field. This field provides vehicle identification information which can be used for per use billing.

Message Type	Description
1	Information Call Request (8.3.1)
2	Information Call Reply (8.3.2)
3	Information Call Data Transfer (10.3.1)
4	Information Call Terminate (8.3.3)

Table 19: Information Call Message Set

10.3.1 Information Call Data Transfer

This message is sent from the SO to the TCU. It provides the data stream for the information that was requested by the TCU.

Octet \ bit	0	1	2	3	4	5	6	7
1…3	Application Header (6)							
4…8	Version (17.4) [OV]							
9…n	Information Type (17.38)							

Table 20: Information Call Data Transfer

11 Remote Vehicle Function Application

11.1 Remote Vehicle Function protocol (Normative)

The remote vehicle function application allows vehicle functions to be controlled remotely by the SO. These functions may include devices such as door lock motors, window motors, trunk release, or modes such as tracking mode, covert mode, etc. The action to remotely control a function may be as a result of the TCU sending a request message to perform a remote vehicle function or a driver's verbal request to perform a remote vehicle function.

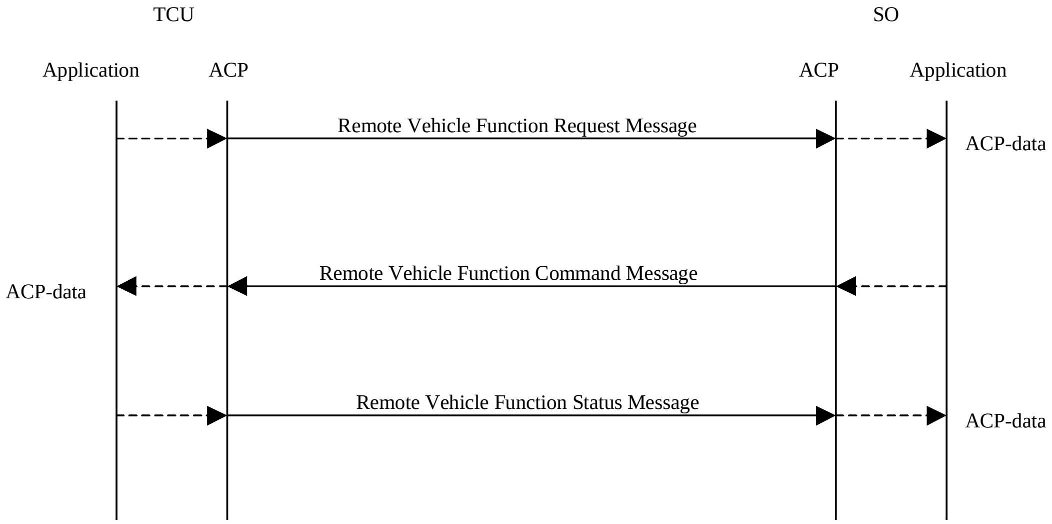
In general, a request is made to the SO by either the TCU or the driver for a specific remote vehicle function to be performed. The SO then initiates the remote vehicle function, assuming a successful authentication.

11.2 Example of Remote Vehicle Function Message Exchanges

A remote vehicle function may be initiated by the TCU or by the SO. When the TCU is the initiator, it is requesting permission to perform the remote vehicle function. When the SO is the initiator, it is commanding the TCU to perform a remote vehicle function at the verbal request of the customer. The following sections illustrate the two methods of initiating a remote vehicle function message exchange.

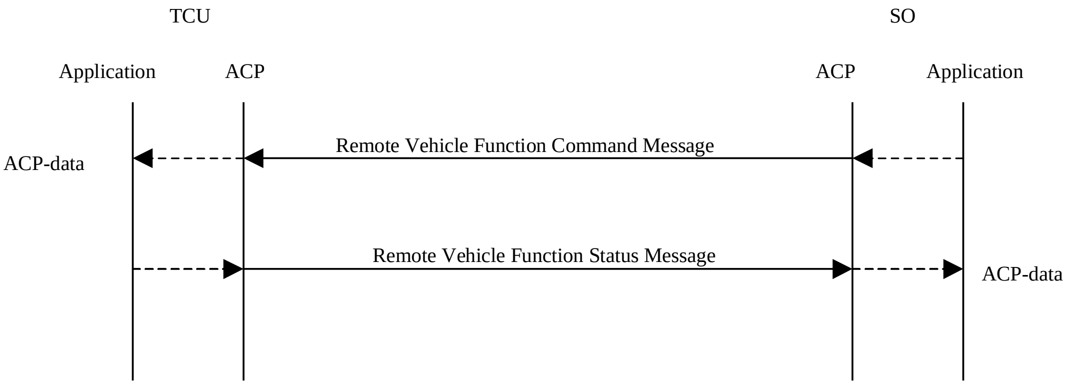
11.2.1 Example of TCU Initiated Remote Vehicle Function Message Exchange

An example of a TCU initiated remote vehicle function is the remote vehicle door unlock function. In the event the driver locks the vehicle keys in the car, he may contact the SO and request that his vehicle doors be unlocked. The SO will ask the driver to press a button on the car. The TCU detects this button press and sends a remote vehicle function request message to ask permission to unlock the vehicle doors. The SO responds with the remote vehicle function command message to unlock the vehicle doors. The TCU sends a remote vehicle function status message to notify the status of the remote vehicle function.


TCU
SO
Application
ACP
ACP
Application
Remote Vehicle Function Request Message
ACP-data
Remote Vehicle Function Command Message
ACP-data
Remote Vehicle Function Status Message
ACP-data

11.2.2 Example of SO Initiated Remote Vehicle Function Message Exchange

The SO may initiate a remote vehicle function per request of the customer or dealership. The remote vehicle function command message can be sent to command the TCU to perform a remote function. The TCU performs the remote function and responds with a remote vehicle function status message to notify the SO of the status of the remote function.


TCU
SO
Application
ACP
ACP
Application
ACP-data
Remote Vehicle Function Command Message
Remote Vehicle Function Status Message
ACP-data

11.3 Remote Vehicle Function Message Set (Normative)

These are the actual messages required to perform a remote vehicle function.

Message Type	Description
1	Remote Vehicle Function Request (11.3.1)
2	Remote Vehicle Function Command (11.3.2)
3	Remote Vehicle Function Status (11.3.3)

11.3.1 Remote Vehicle Function Request

This message is a request from the TCU to perform a remote vehicle function.

Octet \ bit	0	1	2	3	4	5	6	7
1…3	Application Header (6)							
4…8	Version (17.4) [OV]							
9…n-1	Time Stamp (17.29) [O]							
n…m-1	Vehicle Descriptor (17.23) [V]							
m…k-1	Control Function (17.35)							
k…j-1	Function Command (17.36) [V]							

Table 21: Remote Vehicle Function Request

11.3.2 Remote Vehicle Function Command

This message is sent by the SO to the TCU to control a remote vehicle device as a result of a TCU request or customer request.

Octet \ bit	0	1	2	3	4	5	6	7
1...3	Application Header (6)							
4…8	Version (17.4) [OV]							
9…10	Control Function (17.35)							
11…12	Function Command (17.36) [V]							
13..n	Vehicle Descriptor(17.23)							

Table 22: Remote Vehicle Function Command

11.3.3 Remote Vehicle Function Status

This message is sent by the TCU to the SO and represents the status of the remote vehicle device. This message is sent after the TCU receives the Remote Vehicle Function Command from the SO to perform a remote vehicle function and has completed the remote vehicle function.

Octet \ bit	0	1	2	3	4	5	6	7
1…3	Application Header (6)							
4…8	Version (17.4) [OV]							
9…10	Control Function (17.35)							
11…12	Function Status (17.37)							
13..n-1	Vehicle Descriptor (17.23)							
N	Error Element (17.13)							

Table 23: Remote Vehicle Function Status

12 Theft Alarm Application

12.1 Theft Alarm Protocol (Normative)

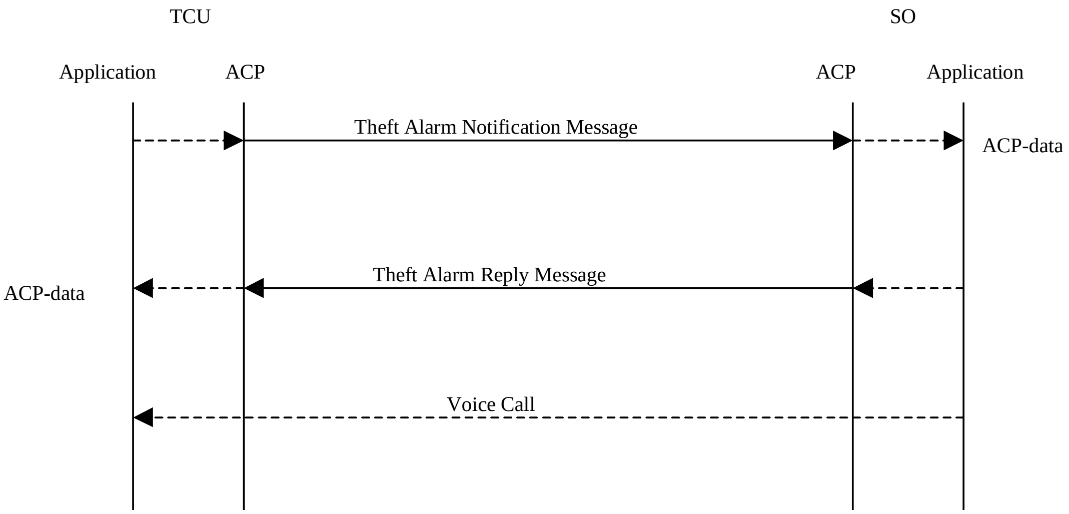
The theft alarm application informs the SO that a vehicle theft alarm has been triggered. The vehicle manufacturer defines the set of conditions that define a theft attempt. This is usually when an atypical startup sequence is encountered. Since this application provides the ability to listen to activity within the vehicle, care must be taken to avoid false alarms to address security and privacy concerns.

12.2 Example of Theft Alarm Message Exchanges

The message exchange between the TCU and SO may vary based on the service agreement between the SO and TCU owner as well as the bearer chosen for message transmittal. If the service agreement provides for vehicle tracking in the event of a stolen vehicle then the vehicle tracking feature is enabled after the theft notification. If the chosen bearer supports concurrent voice and data transmission then a voice call may be initiated by the TCU concurrently with a data transmission. The following are common examples of message exchanges between the SO and TCU.

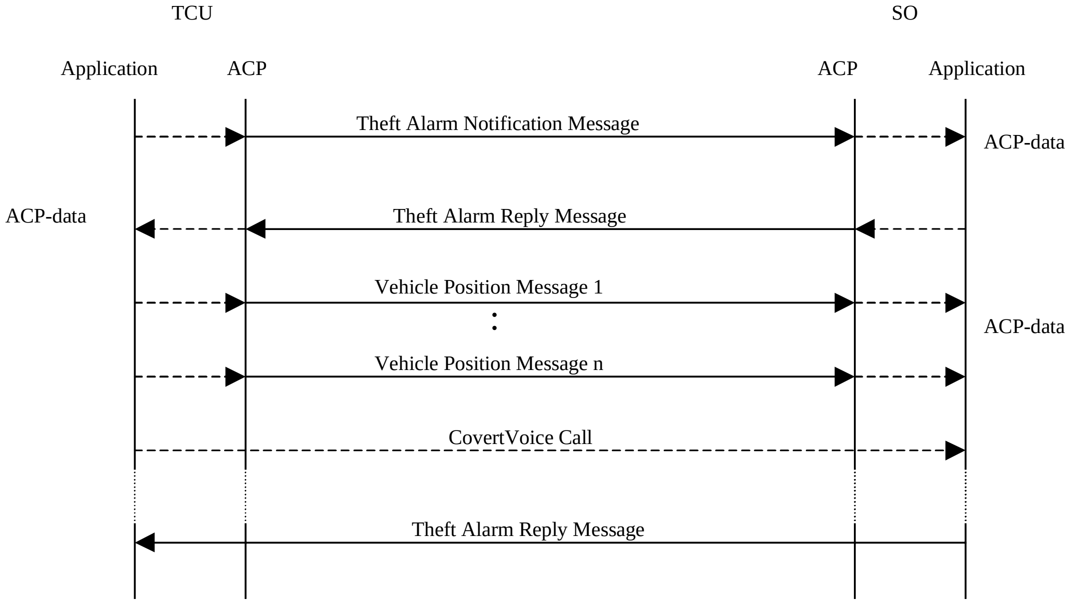
12.2.1 Example of Theft Message With No Tracking

When the TCU determines that a theft is in progress it sends a theft notification to the SO. The SO sends an application level reply to confirm message receipt and to direct the TCU of the next action. The SO then initiates a voice call to the vehicle to determine if the vehicle occupants are authorized operators of the vehicle. If they are authorized then the theft application is terminated. If they are not authorized the SO terminates the voice call and then contacts the police. Below is the message sequence diagram illustrating this message exchange.



12.2.2 Example of Theft Message Followed by Tracking

When the TCU determines that a theft is in progress it sends a theft notification to the SO. The SO sends an application level reply to confirm message receipt and to direct the TCU of the next action. The TCU commences tracking of the vehicle automatically by sending a theft alarm reply message with tracking enabled to the SO. Tracking messages are sent periodically until a theft reply message is sent requesting the tracking function to stop. If the bearer chosen allows concurrent voice and data calls then a voice call is initiated concurrent to the tracking location message. Below is the message sequence diagram illustrating this message exchange.


TCU
SO
Application
ACP
ACP
Application
Theft Alarm Notification Message
ACP-data
ACP-data
Theft Alarm Reply Message
Vehicle Position Message 1
:
ACP-data
Vehicle Position Message n
CovertVoice Call
Theft Alarm Reply Message

12.3 Theft Alarm Message Set (Normative)

These are the actual messages required to accomplish a theft alarm notification. The theft alarm message set is the same as the emergency call message set (8.3). The distinction is made in the Application ID field and the Message Type field of the application header.

Message Type	Description
1	Theft Alarm Notification (8.3.1)
2	Theft Alarm Reply (8.3.2)
3	Vehicle Position Message (8.3.1)

13 Vehicle Tracking Application

13.1 Vehicle Tracking (Normative)

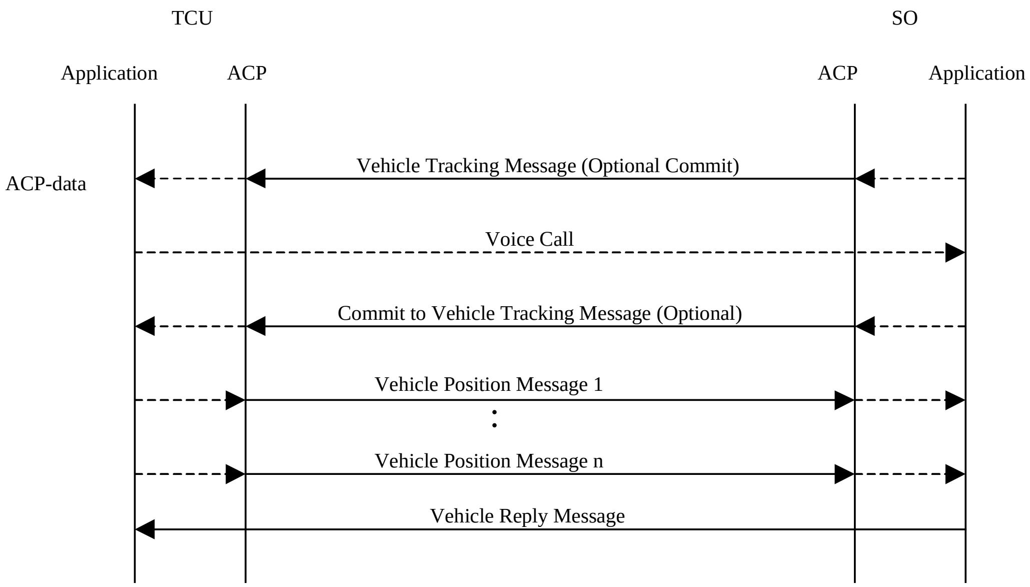
The vehicle tracking application sends vehicle location information to the SO on a periodic basis. This enables the SO to track the vehicle in the event that a vehicle was reported missing. Tracking may automatically be enabled in a SO response of another application. For example, if an emergency call is sent from the TCU to the SO, the SO may respond with an emergency call acknowledged plus vehicle tracking enabled within the same response. This holds true for other applications as well. However, in the event a vehicle owner reports the vehicle missing, the SO must enable the vehicle tracking manually.

In general, the SO is notified of a missing vehicle by the vehicle owner. The SO then sends a vehicle tracking command message to the TCU to enable vehicle tracking. Periodic messages are sent by the TCU to the SO with location information used in the tracking of the vehicle. A vehicle tracking command message can be sent from the SO to the TCU to disable tracking when it is determined that it is no longer needed.

13.2 Vehicle Tracking Message Exchanges

When a missing vehicle is reported, the SO sends a vehicle tracking command to the TCU to enable vehicle tracking. The vehicle may automatically detect a theft based on a set of conditions defined by the vehicle manufacturer. The vehicle may also detect an accident based on various sensor readings throughout the vehicle. The vehicle owner may also place a call to the SO to report a stolen vehicle or help in the recovery of the vehicle under any other circumstances. The TCU then sends the vehicle position message on a periodic basis. The SO sends a position acknowledged message for every position message sent by the TCU. When it is determined that vehicle tracking is no longer needed, the SO sends a tracking message command to the TCU to disable vehicle tracking. This message exchange is illustrated in the message sequence diagram below.

There is an optional commit message to allow for a two step vehicle tracking invocation. This is similar to the Provisioning two step process for the same reasons.


TCU
SO
Application
ACP
ACP
Application
ACP-data
Vehicle Tracking Message (Optional Commit)
Voice Call
Commit to Vehicle Tracking Message (Optional)
Vehicle Position Message 1
Vehicle Position Message n
Vehicle Reply Message

13.3 Vehicle Tracking Message Set (Normative)

These are the actual messages required for vehicle tracking.

Message Type	Description
1	Vehicle Tracking Command (11.3.1)
2	Vehicle Position Message (13.3.2)
3	Vehicle Position Reply (13.3.3)
4	Vehicle Tracking Command with Commit (11.3.1)
5	Commit to Vehicle Tracking Command (11.3.1)

13.3.1 Vehicle Tracking Command Message

This message is sent from the SO to the TCU to enable vehicle tracking. This message is the same as the Remote Vehicle Function Request Message (11.3.1), except with different values for the Control Function, Function Command, and Application and Message Type. The Control Functions (17.35) of interest to the vehicle tracking application are **vehicle tracking, covert mode,** and **microphone**. These functions can be enabled or disabled using this message.

13.3.2 Vehicle Position Message

This message is sent by the TCU to the SO. It is sent by the TCU periodically after vehicle tracking is enabled. This message is the same as the emergency call request message (8.3.1) except with a different Application ID and Message Type. It can also be used to terminate vehicle tracking and covert mode.

13.3.3 Vehicle Position Reply Message

This message is a reply to the vehicle position message above sent by the SO to the TCU. This message is the same as the emergency call reply message (8.3.2) except with a different Application ID and Message Type. It can also be used to terminate vehicle tracking and covert mode.

14 Tele-Diagnostics Application

14.1 Tele-Diagnostics (Normative)

The Tele-Diagnostics application is used to retrieve diagnostic information over a remote wireless connection. This application provides the ability to run and retrieve diagnostic self tests, retrieve diagnostic failure codes, retrieve parameter data, and send diagnostic commands to the TCU. These services can be performed for all modules on the in-vehicle network.

Tele-Diagnostics can be initiated by either the TCU or the SO. During provisioning the TCU may be configured to automatically initiate Tele-Diagnostics when a pre-defined set of criteria have been satisfied. For example, if the emergency call button is stuck, then a message may be initiated by the TCU to send a diagnostic trouble code to the SO indicating a stuck emergency call switch. In the event the vehicle owner reports a problem with the vehicle, the SO can request diagnostic data from the TCU for those diagnostics that are configured to run continuously.

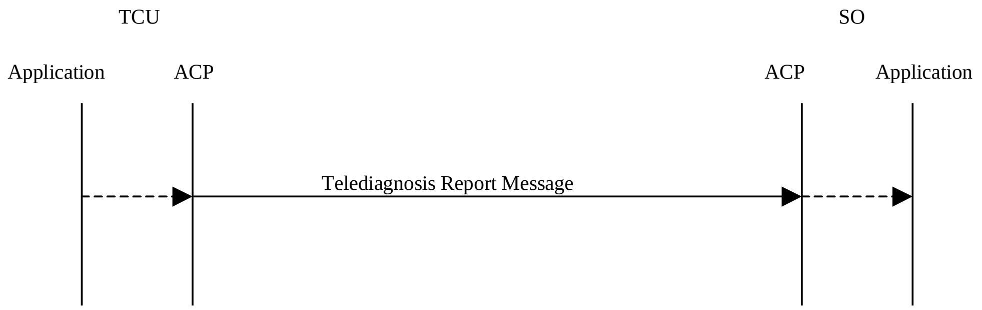
In general, the SO requests diagnostic data from the TCU. The TCU performs the diagnostic service and returns the data to the SO.

14.2 Tele-Diagnostics Message Exchange

Tele-Diagnostics can be initiated by either the TCU or the SO. The first scenario illustrates the message exchange initiated by the TCU. The second scenario illustrates the message exchange initiated by the SO.

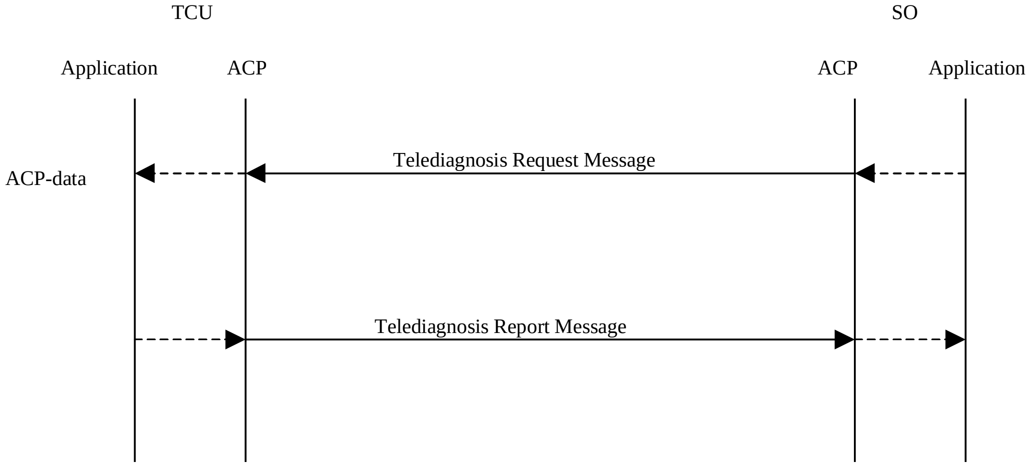
14.2.1 Example of TCU Initiated Tele-Diagnostics Message Exchange

During provisioning the TCU can be configured for Tele-Diagnostics such that when certain predetermined criteria have been satisfied, a Tele-Diagnostics report message is sent from the TCU to the SO to indicate a diagnostic error condition. This is illustrated in the following message sequence diagram.



14.2.2 Example of SO Initiated Tele-Diagnostics Message Exchange

The SO may initiate a call to the TCU when diagnostic information is needed. This is typical when the dealership requires diagnostic information from the vehicle to perform remote diagnosis. The SO sends a Tele-Diagnostics request message to the TCU. The TCU responds with the Tele-Diagnostics report message containing the requested diagnostic information. This is illustrated in the following message sequence diagram.



14.3 Tele-Diagnostics Message Set (Normative)

These are the actual messages required for Tele-Diagnostics.

Message Type	Description
1	Tele-Diagnostics Request (14.3.1)
2	Tele-Diagnostics Report (14.3.2)

14.3.1 Tele-Diagnostics Request Message

This message is sent by the SO to the TCU to request diagnostic services. This message is used when diagnostic information is requested by the dealership to diagnose vehicle problems. The Diagnostic Data Service and Source are used to request a specific diagnostic test to be performed.

Octet \ bit	0	1	2	3	4	5	6	7
1…3	Application Header (6)							
4…8	Version (17.4) [OV]							
9	Diagnostic Data Service (17.40)							
10	Diagnostic Data Source (17.41)							
11…k-1	Diagnostic Data (17.42) [V]							

Table 24: Tele-Diagnostics Request Message

14.3.2 Tele-Diagnostics Report Message

This message is sent by the TCU to the SO. It is used to report diagnostic information in the event the TCU has detected a vehicle diagnostic failure. It is also used to report diagnostic data requested by the SO via the Tele-Diagnostics request message.

Octet \ bit	0	1	2	3	4	5	6	7
1…3	Application Header (6)							
4…8	Version (17.4) [OV]							
9…n-1	Time Stamp (17.25)							
n…m-1	Vehicle Descriptor (17.23) [V]							
m…l-1	Location (17.1) [V]							
L	Error Element (17.13)							
K	Diagnostic Data Service (17.40)							
k+1	Diagnostic Data Source (17.41)							
k+2…h-1	Diagnostic Data (17.42) [V]							

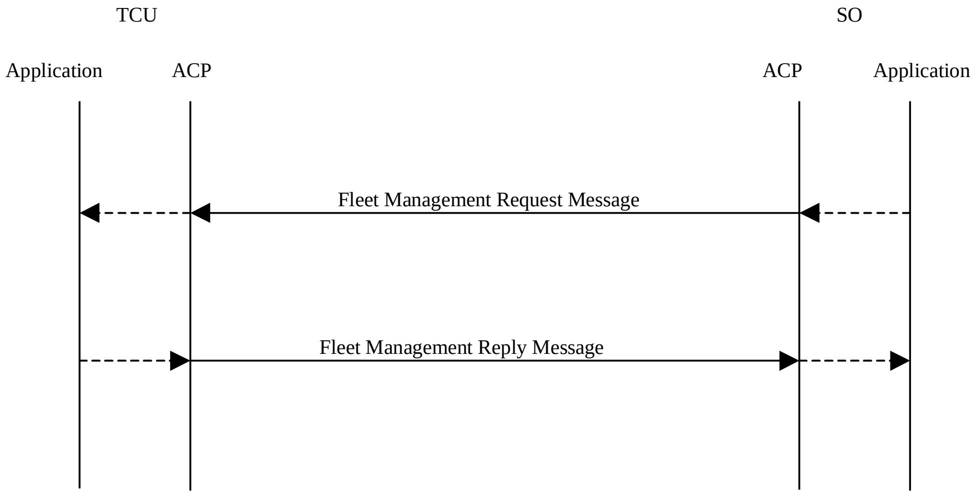
Table 25: Tele-Diagnostics Request Message

15 ACP Fleet Management Application

The fleet management application is used to obtain operational data on the use of a group of vehicles, typically within one company or organization.

15.1 Example Fleet Management Message Exchange

The following is an example of a fleet management message exchange. A Fleet Management Request message is sent from the SO to the TCU. The TCU then sends a Fleet Management Reply message back to the SO with the Fleet Management information. Below is the message sequence illustrating this message exchange.


TCU
SO
Application
ACP
ACP
Application
Fleet Management Request Message
Fleet Management Reply Message

Message Type	Description
1	Fleet Management Request
2	Fleet Management Reply

Table 26: Fleet Management Message Set

15.2 Fleet Management Request Message

This message is sent by the SO to the TCU to request fleet management data.

Octet \ bit	0	1	2	3	4	5	6	7
1-3	Application Header (6)							
4…8	Version (17.4) [OV]							
9	Vehicle Fleet Block Control Flag (15.2.1)							
10	Vehicle Fleet Event Flag (15.2.2) [O]							
10…m-1	Vehicle Descriptor (17.23) [OV]							

Table 27: Fleet Management Request Message

15.2.1 Vehicle Fleet Block Control Flag

Bit	Definition
0	1=Another control flag follows 0=Last control flag
1	1=Speed Profile Element and Current Speed Requested 0=No request
2	1=Fuel Consumption Element and Requested 0=No request
3	1=Engine Speed Element and Current Engine Speed Requested 0=No request
4	1=Coolant Profile Element and Current Coolant Temperature Requested 0=No request
5	1=Oil Temperature and Pressure Element and Current Oil Temperature and Pressure Requested 0=No request
6	1=First/Last Location Element Requested 0=No request
7	1=Service Profile Element Requested 0=No request

Table 28: Vehicle Fleet Block Control Flag Encoding

15.2.2 Vehicle Fleet Event Flag

Bit	Definition
0	1=Another control flag follows 0=Last control flag
1	1=Current Odometer Reading Requested 0=No request
2	1=Current Location Requested 0=No request
3	Reserved, set to 0
4	Reserved, set to 0
5	Reserved, set to 0
6	Reserved, set to 0
7	Reserved, set to 0

Table 29: Vehicle Fleet Event Flag Encoding

15.3 Fleet Management Reply Message

This message is sent by the TCU in response to a SO requesting fleet management data. It can be sent by the TCU without a prior Fleet Management Request message if the TCU is configured to automatically send data based on a configured pre-condition.

Fleet Management data collection can be configured either as the profile method or the raw data sample method. If the sample method was defined, then the data type returned is defined in section (15.16) for speed, fuel, engine, coolant, and oil data elements.

Octet \ bit	0	1	2	3	4	5	6	7
1-3	Application Header (6)							
4…8	Version (17.4) [OV]							
9	Reserved							
10	Number of Vehicle Fleet Blocks (15.4)							
110…n-1	Vehicle Fleet Block (15.5)							
N	Error Element (17.13)							
N+1…m-1	Vehicle Descriptor (17.23) [OV]							
m…k-1	Time Stamp (17.29) [O]							
k…j-1	Location (17.14) [OV]							
j…p-1	Service Profile (15.17) [OV]							

Table 30: Fleet Management Reply Message

The *Application ID* field in the ACP transport header indicates fleet management.

15.4 Number of Vehicle Fleet Block

This is the number of Vehicle Fleet Blocks in this message.

15.5 Vehicle Fleet Block

The requested fields shall be ordered as shown below.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier		More flag	Length				
2-5	Time Stamp (17.29)							
6-9	Reserved							
10	Temperature Flag (15.5.11)		Volume Flag (15.5.10)		Distance Flag (15.5.9)		Pressure Flag (15.5.12)	
11	Reserved	Reserved	Reserved	Reserved	Most Significant bits of Odometer			
12,13	Current odometer reading (15.5.2)							
14	Odometer distance (15.5.3)							
15	Current Speed (15.5.4) [O]							
16	Current Engine Speed (15.5.5) [O]							
17	Current Coolant Temperature (15.5.6) [O]							
18	Current Oil Temperature (15.5.7) [O]							
19..n-1	Current Oil Pressure (15.5.8) [O]							
N…m-1	Speed profile element (15.6) [OV]							
M…k-1	Fuel consumption element (15.8) [OV]							
K…j-1	Engine speed element (15.10) [OV]							
J…i-1	Coolant profile element (15.12) [OV]							
H…s-1	First/Last vehicle position element (15.14) [O]							
S…t-1	Oil temperature and oil pressure element (15.15) [O]							

15.5.1 Time Stamp

The starting time at which data for this fleet block was accumulated.

15.5.2 Current Odometer Reading

The current value of the odometer when the fleet block is transmitted. The units are defined in the *Distance Flag. This field allows up to 1 million miles.*

15.5.3 Odometer Distance

The distance traveled since time indicated by field *Time Stamp*. The units are defined in the *Distance Flag.*

15.5.4 Current Speed

The current speed value when the vehicle fleet block is transmitted. The units are defined in the *Distance Flag.*

15.5.5 Current Engine RPM

The current speed value when the vehicle fleet block is transmitted. The units are revolutions per minute.

15.5.6 Current Coolant Temperature

The current coolant temperature value when the vehicle fleet block is transmitted. The units are defined in the *Temperature Flag*. Negative coolant temperatures are reported as 0. This is a temporary condition when the engine is started in cold climates.

15.5.7 Current Oil Temperature

The current oil temperature value when the vehicle fleet block is transmitted. The units are defined in the *Temperature Flag.*

15.5.8 Current Oil Pressure

The current oil pressure value when the vehicle fleet block is transmitted. The units are defined in the *Pressure Flag.*

15.5.9 Distance Flag

Value	Definition
0	Units are not defined.
1	The units are kilometers
2	The units are miles

Table 31: DistanceFlag encoding

15.5.10 Volume Flag

Value	Definition
0	Units are not defined
1	The units are liters
2	The units are gallons

Table 32: VolumeFlag encoding

15.5.11 Temperature Flag

Value	Definition
0	Units are not defined
1	The units are degrees Centigrade
2	The units are degrees Fahrenheit

Table 33: TemperatureFlag encoding

15.5.12 Pressure Flag

Value	Definition
0	Units are not defined
1	The units are in TBD
2	The units are in Pounds per Square Inch (PSI)

Table 34: PressureFlag encoding

15.6 Speed Profile Element

Octet \ bit	0	1	2	3	4	5	6	7
1	3 (Extended IE)		IE Identifier					
2	More Flag	Length						
3	Average speed (15.6.3)							
4	Maximum speed (15.6.4)							
5-6	Moving time (15.6.5)							
7…n-1	Percentage speed element (15.7)							
n…m-1	Nth percentage speed element (15.7)							

15.6.1 IE Identifier

IE Identifier value of 4 identifies this element as a *speed profile* element.

15.6.2 Length

Since the number of *percentage speed elements* is variable, this indicates the size of this element and thus the number of *percentage speed* elements. The size of each *percentage speed* elements is 2 octets

15.6.3 Average Speed

Average speed when vehicle is moving. Stationary time is not included in the average. This defined as units/hour where units are as defined in section 15.5.9.

15.6.4 Maximum Speed

This is the maximum speed attained in time period. This defined as units/hour where units are as defined in section 15.5.9. The maximum value is 255 units/hour.

15.6.5 Moving Time

The time in minutes while engine is on that the vehicle speed is non-zero.

15.7 Percentage Speed Element

Octet \ bit	0	1	2	3	4	5	6	7
1	Speed (15.7.1)							
2	Percentage speed (15.7.2)							

The speed profile element is used to build up a table of speeds and percentage of distance traveled at those speeds. Each table entry in speed profile element is defined by the percentage speed element

15.7.1 Speed

The field *speed* specifies the range of speed in this table. If no prior *percentage speed element* exists then the range is from 0 to the value of *speed*. If a prior *percentage speed* element exists (n-1), the field *speed* specifies the speed in excess of the n-1 *speed* field and less than or equal to the current *speed* field.

15.7.2 Percentage Speed

This is the percentage total distance over which the range of *speed* occurred.

15.8 Fuel Consumption Element

Octet \ bit	0	1	2	3	4	5	6	7
1	3 (Extended IE)		IE Identifier					
2	More Flag	Length						
3…n-1	Percentage fuel consumption element (15.9)							
n…m-1	Nth percentage fuel consumption element (15.9)							

15.8.1 IE Identifier

IE Identifier value of 5 (binary data) identifies this element as a *fuel consumption* element.

15.8.2 Length

Since the number of *percentage fuel consumption* is variable, this indicates the size of this element and thus the number of percentage speed elements.

15.9 Percentage Fuel Consumption Element

Octet \ bit	0	1	2	3	4	5	6	7
1-2	Fuel consumed (15.7.1)							
3-4	Distance-2 (15.7.2)							

This element specifies the fuel consumed during several periods of time. Each pair of fields F*uel consumed* and *distance* specify a period and the fuel consumed during that period. The field *length* can be used to determine the length of this element.

15.9.1 Fuel Consumed

This is the amount of fuel consumed. This field has a range from 0 to 65536. The units for this field are defined in section 15.5.10

15.9.2 Distance

This is the distance traveled while measuring the above fuel consumed. This field has a range from 0 to 65536. The units for this field are defined in section 15.5.9.

15.10 Engine Speed Element

RPM is defined as Revolutions per Minute.

Octet \ bit	0	1	2	3	4	5	6	7
1	3 (Extended IE)		IE Identifier					
2	More Flag	Length						
3	Average RPM (15.9)							
4	Maximum RPM (15.10.3)							
5	Current RPM (15.10.4)							
6	Number of engine starts (15.10.5)							
7,8	Engine running time (15.10.6)							
9…n-1	Percentage RPM element (15.11)							
n…m-1	Nth percentage RPM element (15.11)							

Table 35: Engine Speed element encoding

15.10.1 IE Identifier

IE Identifier value of 6 (binary data) identifies this element as an *engine speed* element.

15.10.2 Length

Since the number of *percentage RPM elements* is variable, this indicates the size of this element and thus the number of *percentage RPM elements*.

15.10.3 Maximum RPM

This is the maximum RPM the vehicle attained in the snapshot. The maximum value is 255. The units are in hundreds of RPM. The value of 255 represents 25500 RPM.

15.10.4 Current RPM

The current RPM of this vehicle when the flock block is transmitted. The maximum value is 255. The units are in hundreds of RPM. The value of 255 represents 25500 RPM.

15.10.5 Number of Engine Starts

The number of engine starts for this fleet block.

15.10.6 Engine Running Time

The amount of time the engine has been running for this fleet block. The units are in hours. The value of 65535 is to be interpreted as an overflow and the actual running time can not be determined.

15.11 Percentage RPM Element

Octet \ bit	0	1	2	3	4	5	6	7
1	RPM (15.11.1)							
2	Percentage RPM (15.11.2)							

Table 36: Percentage RPM encoding

The RPM profile element is used to build up a table of RPMs and percentage of distance traveled at those RPMs. Each table entry in RPM profile element is defined by the percentage RPM element

15.11.1 RPM

The field *RPM* specifies the range of RPM in this table. If no prior *percentage RPM element* exists then the range is from 0 to the value of *RPM.* If a prior *percentage RPM* element exists (n-1), the field *RPM* specifies the RPM in excess of the (n-1) *RPM* field and less than or equal to the current *RPM* field.

15.11.2 Percentage RPM

This is the percentage total distance over which the range of RPM occurred.

15.12 Coolant Profile Element

The units for temperature can be either Centigrade or Fahrenheit as defined in T*emperature Flag* section.15.5.11 It is up to the application to determine the units and use them consistently. Negative coolant temperatures are reported as 0. This is a temporary condition when the engine is started in cold climates.

Octet \ bit	0	1	2	3	4	5	6	7
1	3 (Extended IE)		IE Identifier					
2	More Flag	Length						
3	Average coolant temperature (15.12.3)							
4	Maximum coolant temperature (15.12.4)							
5	Current coolant temperature (15.12.5)							
6…n-1	Percentage coolant element (15.13.2)							
n…m-1	Nth percentage coolant element (15.13.2)							

Table 37: Colant Profile element encoding

15.12.1 IE Identifier

IE Identifier value of 7 (binary data) identifies this element as a *coolant profile* element.

15.12.2 Length

Since the number of *percentage coolant elements* is variable, this indicates the size of this element and thus the number of *percentage coolant elements*.

15.12.3 Average Coolant Temperature

This is the average coolant temperature for the vehicle for this fleet data block. The maximum value is 255. The units are defined in (15.5.11).

15.12.4 Maximum Coolant Temperature

This is the maximum coolant temperature the vehicle attained for this fleet data block. The maximum value is 255. The units are defined in (15.5.11).

15.12.5 Current Coolant Temperature

This is the current coolant temperature for the vehicle when this fleet data block is transmitted. The maximum value is 255. The units are defined in (15.5.11).

15.13 Percentage Coolant Element

Octet \ bit	0	1	2	3	4	5	6	7
1	Coolant temperature (15.13.1)							
2	Percentage coolant temperature (15.13.2)							

The coolant profile element is used to build up a table of coolant temperatures and percentage of distance traveled at those coolant temperatures. Each table entry in coolant profile element is defined by the percentage coolant temperature element

15.13.1 Coolant Temperature

The field *coolant temperature* specifies the range of coolant in this table. If no prior *percentage coolant temperature element* exists then the range is from 0 to the value of *coolant temperature.* If a prior *percentage coolant temperature* element exists (n-1), the field *coolant temperature* specifies the coolant in excess of the n-1 *coolant temperature* field and less than or equal to the current *coolant temperature* field.

15.13.2 Percentage Coolant

This is the percentage total time over which the range of coolant temperature occurred.

15.14 First/Last Location Element

Octet \ bit	0	1	2	3	4	5	6	7
1	3 (Extended IE)		IE Identifier					
2	More Flag	Length						
3…n-1	First location element (15.14.3)							
n…m-1	Last location element (15.14.4)							

15.14.1 IE Identifier

IE Identifier value of 8 (binary data) identifies this element as a first/last location element.

15.14.2 Length

Since the number of *location elements* is variable, this indicates the size of this element and thus the number of percentage speed elements.

15.14.3 First Location Element

The start location for this fleet block when started accumulating data. The units are defined in section 17.16.

15.14.4 Last Location Element

The location for this fleet block when the fleet block was transmitted. The units are defined in section 17.16.

15.15 Oil Temperature and Pressure Profile Element

The units for oil temperature can be either Centigrade or Fahrenheit as defined in T*emperature Flag* section.15.5.11. The units for oil pressure are defined in the Pressure Flag 15.5.12. It is up to the application to determine the units and use them consistently.

Octet \ bit	0	1	2	3	4	5	6	7
1	3 (Extended IE)		IE Identifier					
2	More Flag	Length						
3	Average oil temperature (15.15.3)							
4	Maximum oil temperature (15.15.4)							
5	Current oil temperature (15.15.5)							
6	Average oil pressure (15.15.6)							
7	Maximum oil pressure (15.15.7)							
8	Current oil pressure (15.15.8)							

Table 38: Oil profile element encoding

15.15.1 IE Identifier

IE Identifier value of 9 (binary data) identifies this element as a *oil temperature and pressure profile* element.

15.15.2 Length

Since the number of *oil temperature and pressure* is fixed, this indicates the size of this element to be compatible with the other profile elements defined in this section.

15.15.3 Average Oil Temperature

This is the average oil temperature for the vehicle for this fleet data block. The maximum value is 255. The units for oil pressure are defined in the Temperature Flag 15.5.11

15.15.4 Maximum Oil Temperature

This is the maximum oil temperature the vehicle attained for this fleet data block. The maximum value is 255. The units for oil pressure are defined in the Temperature Flag 15.5.11.

15.15.5 Current Oil Temperature

This is the current oil temperature for the vehicle when this fleet data block is transmitted. The maximum value is 255. The units are defined in the Temperature Flag 15.5.11

15.15.6 Average Oil Pressure

This is the average oil pressure for the vehicle for this fleet data block. The maximum value is 255. The units for oil pressure are defined in the Pressure Flag 15.5.12.

15.15.7 Maximum Oil Pressure

This is the maximum oil pressure the vehicle attained for this fleet data block. The maximum value is 255. The units for oil pressure are defined in the Pressure Flag 15.5.12.

15.15.8 Current Oil Pressure

This is the current oil pressure for the vehicle when this fleet data block is transmitted. The maximum value is 255. The units for oil pressure are defined in the Pressure Flag 15.5.12.

15.16 Raw Data Element

Octet \ bit	0	1	2	3	4	5	6	7
1	3 (Extended IE)		IE Identifier					
2	More Flag	Length						
3..n	Data							

The IE Identifier is defined by the following table and the data units are defined by the data unit flags.

Value	Definition
10	Units are not defined
11	Speed
12	Engine RPM
13	Coolant temperature
14	Oil temperature
15	Oil pressure

15.17 Service Profile

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier		More flag	Length				
2	Fleet Service (15.17.1)							
3	Data Units (15.17.2)							
4…n	Service Data							

15.17.1 Fleet Service

Value	Definition	Service Data Content
0	Reserved	
1	Current odometer reading	24 bits

2	Current location	Location encoding (17.16)
3-255	Not defined	

15.17.2 Data Units

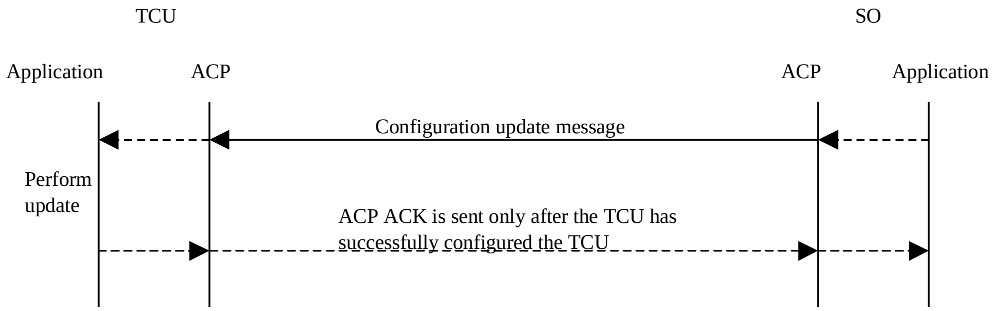
Value	Definition
0	Units not defined
1	Miles
2	Kilometers
3	Meters
4	Liters
5	Gallons
6	Centigrade
7	Fahrenheit
8-255	Not Defined

16 ACP Application Configuration

16.1 General Configuration Protocol

16.1.1 Example of very simple SO Initiated Configuration

This example assumes a reliable media for Configuration in the TCU. The application in the TCU does not have a separate confirmation message, but uses the transport acknowledgement to inform the SO of successful provisioning.


TCU
SO
Application
ACP
ACP
Application
Configuration update message
Perform update
ACP ACK is sent only after the TCU has successfully configured the TCU

16.1.2 Protocol (Normative)

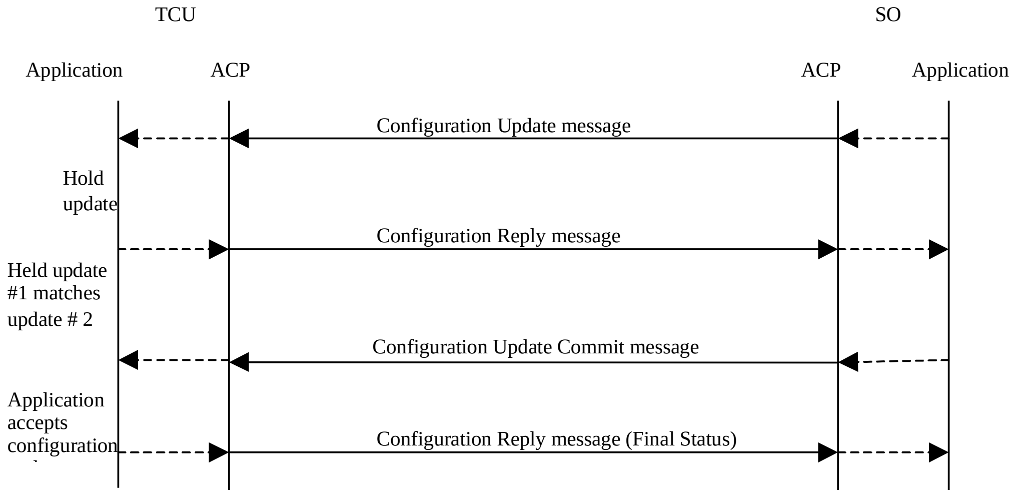
These are messages that allow the service operator (SO) and the TCU in the vehicle to configure applications. This allows for the establishment of features and services by application. Parameter updates are usually done within an application. If the service operator wants to sign up a customer when the service is 1st used, then more information must be supplied as part of the application, information the customer may not readily have at time of use.

The *Update message* allows the SO to begin the update sequence whereas the *Update Request message* allows the TCU to begin the update sequence. Since the SO will not usually know when the TCU is available for the *Update message*. Updates are usually initiated by the SO. An optional capability exists that allows the TCU to initiate the configuration exchange.

16.1.3 Example of SO Initiated Provision Message Exchange

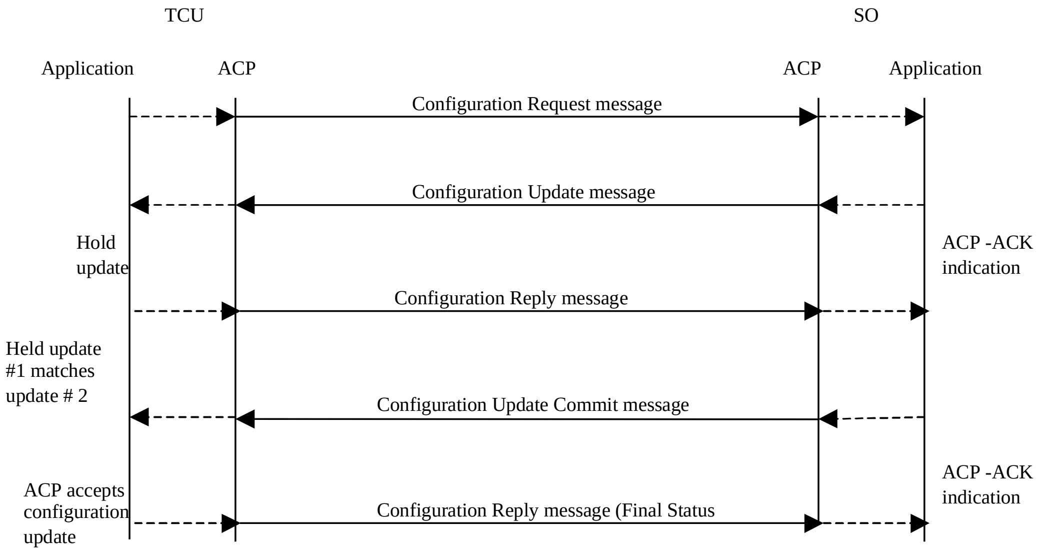
This is an example of the initial message exchange that is necessary for the SO to configure the TCU with the appropriate information to collect data [e.g .Speed Profiles].

The *Update Commit Message* is available for those applications that want a more robust configuration mechanism This is especially important for SMS when realizing that when the SO sends a message to the TCU, the TCU may be turned off and the SMS message is queued at the SMSC. When C*onfiguration Update Commit message* is used, then the sender must receive some acknowledgement that the *configuration* was successful.


TCU
SO
Application
ACP
ACP
Application
Configuration Update message
Hold update
Configuration Reply message
Held update #1 matches update # 2
Configuration Update Commit message
Application accepts configuration
Configuration Reply message (Final Status)

16.1.4 Example of TCU Initiated Message Exchange

This is an example of a TCU requesting an update of Fleet Management configuration data from the SO. An example of this is the TCU's service agreement is expiring and the TCU's owner wants to renew the agreement. The SO may choose to renew or not renew.


TCU
SO
Application
ACP
ACP
Application
Configuration Request message
Configuration Update message
Hold update
ACP -ACK indication
Configuration Reply message
Held update #1 matches update # 2
Configuration Update Commit message
ACP -ACK indication
ACP accepts configuration update
Configuration Reply message (Final Status

16.2 Fleet Management Application Configuration

16.2.1 Fleet Management Configuration Message Set (Normative)

These are the messages required to accomplish a Fleet Management configuration call.

There are two methods in which to collect fleet management data. The first is by profiles. Profiles can be configured to collect and process the data before transmission to the service operator. The second is a collection of samples of raw data to be processed at the service operator. The method of data collection is specified in ControlFlag2 (17.3). If the sample method of collection is selected, then Configuration Flag (16.2.1.1.2) is sufficient to define what is to be collected.

Message Type	Description
1	Fleet Management Configuration Update Message (0)
2	Optional Fleet Management Configuration Update Commit *M*essage (16.2.1.2)
3	Fleet Management Configuration Reply Message (16.2.1.3)
4	Optional Fleet Management Configuration Reply Commit Message #2 (16.2.1.3)
5	Fleet Management Configuration Request Message (16.2.1.4)
6	Fleet Management Configuration Status Message (16.2.1.5)
7	Fleet Management Configuration Edit Message (16.2.1.6)

16.2.1.1 Fleet Management Configuration Update message

This message is sent from SO to TCU to update fleet management information, enable or disable fleet management data collection, and define when fleet management data is transmitted in the TCU. The control flags indicate the presence of optional fields. The included optional fields shall be ordered as shown below. Note that a length field of zero indicates no data is present for that element.

In the Service Profile Configuration, the odometer is the only configurable item. When configured, the current Odometer value will be sent ever time a multiple of the configured odometer value is reached. For example, if the configured odometer value was 10,000 Kilometers, then the current odometer value would be sent at odometer=10,000; odometer=20,000; odometer=30,000; …until the configured odometer value is set to 0.

Octet \ bit	0	1	2	3	4	5	6	7
1...3	Application header (6)							
4…8	Version (17.4) [OV]							
9	More Flag	Target Application ID (4.1)						
10	ApplFlag1 (17.1)		ControlFlag1(17.2)					
11	ControlFlag2 (17.3)							
12	Reserved							
13-16	Start Time (17.5) [O]							
17-20	End Time (17.6) [O]							
21-24	Grace Time (17.6) [O]							
25	More Flag	Number of Samples (16.2.1.1.3)[O]						
26..n-1	Vehicle Descriptor (17.23) [OV]							
n	TCU Descriptor Element (17.8) [OV]							
z	Configuration Flag (16.2.1.1.2)							
u	Length of Speed Profile Configuration (16.2.1.1.4) [O]							
m..p-1	Speed Profile Configuration (16.2.1.1.5) [OV]							
P	Length of Fuel Consumption Profile Configuration (16.2.1.1.6) [O]							
s..r-1	Fuel Consumption Profile Configuration (16.2.1.1.7) [OV]							
T	Length of Engine Speed Profile Configuration (16.2.1.1.8) [O]							
j..k-1	Engine Speed Configuration (16.2.1.1.9) [OV]							
k	Length of Coolant Configuration (16.2.1.1.10) [O]							
d..w-1	Coolant Profile Configuration (16.2.1.1.11) [OV]							
w..a-1	Oil Temperature and Pressure Configuration (16.2.1.1.13) [OV]							
a..e-1	Service Profile Configuration (15.17) [OV]							
e..f-1	Trigger Conditions (16.2.1.1.14) [O]							
f	Length of Transmit Times (16.2.1.1.14.1) [O]							
r..g-1	Transmit Times (16.2.1.1.14.2) [OV]							
g,h	Odometer Trigger Value (16.2.1.1.14.3) [O]							
h..v-1	Location Trigger (17.14) [O]							

Table 39: Fleet Management update message

16.2.1.1.1 Target Application ID

This field takes on the same values as application ID (4.1). It is used to identify the application that is to be provisioned. The Provisioning and Configuration application Ids should not be used with this message.

16.2.1.1.2 Configuration Flag

This is defined as a 8 bit field with bit 0 on the left and bit 7 on the right.

Bit number	Definition
0	1=Another configuration flag follows 0=Last configuration flag
1	1=Speed Profile length and configuration is present 0=Speed Profile length and configuration is not present
2	1=Fuel Consumption Profile length and configuration is present 0=Fuel Consumption Profile length and configuration is not present
3	1=Engine Speed Profile length and configuration is present 0=Engine Speed Profile length and configuration is not present
4	1=Coolant Profile length and configuration is present 0= Coolant Profile length and configuration is not present
5	1=Oil Temperature and Pressure Configuration is present *0*= Oil Temperature and Pressure Configuration is not present
6	1=Service Profile is present *0*=Service Profile is not present
7	1=Trigger Conditions is present *0*=Trigger Conditions is not present

Table 40: Configuration Flag Values

16.2.1.1.3 Number of Samples

This is the number of samples to be taken between the start time and end time.

16.2.1.1.4 Length of Speed Profile Configuration

Octet \ bit	0	1	2	3	4	5	6	7
1	3 (Extended IE)		IE Identifier					
2	More Flag	Length						

16.2.1.1.4.1 IE Identifier

IE Identifier value of 2 (binary data) identifies this element as Length of Speed Profile Configuration element.

16.2.1.1.4.2 Length

This indicates the size of this the Speed Profile Configuration.

16.2.1.1.5 Speed Profile Configuration

This field is used to configure the Speed Profile Element. The field is a list of Speeds (15.7.1) used to configure the Percentage Speed Element (15.7) Speed field.

16.2.1.1.6 Length of Fuel Consumption Profile Configuration

Octet \ bit	0	1	2	3	4	5	6	7
1	3 (Extended IE)		IE Identifier					
2	More Flag	Length						

16.2.1.1.6.1 IE Identifier

IE Identifier value of 4 (binary data) identifies this element as Length of Fuel Consumption Profile Configuration element.

16.2.1.1.6.2 Length

This indicates the size of this the Fuel Consumption Profile Configuration.

16.2.1.1.7 Fuel Consumption Profile Configuration

This field is used to configure the Fuel Consumption Profile Element. The field is a list of Fuel Consumed (15.9.1) used to configure Percentage Fuel Consumption Element (15.9) fuel consumed field.

16.2.1.1.8 Length of Engine Speed Profile Configuration

Octet \ bit	0	1	2	3	4	5	6	7
1	3 (Extended IE)		IE Identifier					
2	More Flag	Length						

16.2.1.1.8.1 IE Identifier

IE Identifier value of 6 (binary data) identifies this element as Length of Engine Speed Profile Configuration element.

16.2.1.1.8.2 Length

This indicates the size of this the Engine Speed Profile Configuration.

16.2.1.1.9 Engine Speed Profile Configuration

This field is used to configure the Engine Speed Profile Element. The field is a list of RPM (15.11.1) used to configure Percentage Engine Speed Element (15.10) RPM field.

16.2.1.1.10 Length of Coolant Profile Configuration

Octet \ bit	0	1	2	3	4	5	6	7
1	3 (Extended IE)		IE Identifier					
2	More Flag	Length						

16.2.1.1.10.1 IE Identifier

IE Identifier value of 10 (binary data) identifies this element as Length of Coolant Profile Configuration element.

16.2.1.1.10.2 Length

This indicates the size of this the Coolant Profile Configuration.

16.2.1.1.11 Coolant Profile Configuration

This field is used to configure the Coolant Profile Element. The field is a list of Coolant Temperatures (15.13.1) used to configure Percentage Coolant Element (15.13) Coolant Temperature field.

16.2.1.1.12 Length of Oil Temperature and Pressure Profile Configuration

Octet \ bit	0	1	2	3	4	5	6	7
1	3 (Extended IE)		IE Identifier					
2	More Flag	Length						

16.2.1.1.12.1 IE Identifier

IE Identifier value of 12 (binary data) identifies this element as Length of Oil Temperature and Pressure Profile Configuration element.

16.2.1.1.12.2 Length

This indicates the size of this the Oil Temperature and Pressure Profile Configuration.

16.2.1.1.13 Oil Temperature and Pressure Profile Configuration

This field is used to configure the Oil Temperature and Pressure t Profile Element. The field is a list of Oil Temperature and Pressure Temperatures (15.15) used to configure Percentage Coolant Element (15.15) Coolant Temperature field.

16.2.1.1.14 Trigger Conditions

This is defined as a 8 bit field with bit 0 on the left and bit 7 on the right.

Bit number	Definition
0	1=Another Trigger flag follows 0=Last Trigger flag
1	1=The Fleet Data Block is transmitted as specified by the list of transmit times. 0=No Periodic update is requested.
2	1=The Fleet Data Block is transmitted after the Odometer value is exceeded. 0=No odometer trigger is present.
3	1=The Fleet Data Block is transmitted when the Trigger Location is reached. 0=No Trigger Location is present.
4	1=The Fleet Data Block is transmitted when the engine is started. 0=No engine start trigger is requested.
5	1=The Fleet Data Block is transmitted when the engine is turned off. 0=No engine turned off trigger is requested.
6	Reserved and set to 0
7	Reserved and set to 0

16.2.1.1.14.1 Length of Transmit Times

Octet \ bit	0	1	2	3	4	5	6	7
1	3 (Extended IE)		IE Identifier					
2	More Flag	Length						

16.2.1.1.14.1.1 IE Identifier

IE Identifier value of 10 (binary data) identifies this element as Length of Speed Profile Configuration element.

16.2.1.1.14.1.2 Length

This indicates the size of this the Speed Profile Configuration.

16.2.1.1.14.2 List of Transmit Time

This is a list of absolute times to transmit the Fleet Data Block. The units are defined in section 17.29.

16.2.1.1.14.3 Odometer Trigger Value

This is the distance traveled while measuring the above fuel consumed. This field has a range from 0 to 65536. The units for this field are defined in section 15.5.9

16.2.1.1.14.4 Location Trigger Value

This is the location when reached will cause the Fleet Data Block to be transmitted. Location is defined in section 17.16.

16.2.1.2 Fleet Management Configuration Update Commit message

The format of this message is exactly the same as *Fleet Management Configuration Update Message*. However, the value of *ApplFlag1* can differ based on the *Fleet Management Configuration Reply Message* received by the SO. Thus when a system uses this message they are deferring the Fleet Management configuration until the SO knows the TCU can perform the requested configuration action. This is most important when using SMS and using long time to live for SMS messages.

16.2.1.3 Fleet Management Configuration Reply Message

The message is similar to the *Fleet Management Configuration Update* Message except the configuration elements are not present.

Octet \ Bit	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1…3	Application header (6)							
4…8	Version (17.4) [OV]							
9	Reserved							
10	More Flag	Target Application ID (4.1)						
11	ApplFlag1 (17.1)		ControlFlag1(17.2)					
12	StatusFlag1 (17.10)		TCU Response Flag (17.11)		Reserved, set to 0			
13-16	Start Time (17.5) [O]							
17-20	End Time (17.6)[O]							
21	Error element (17.13)							
22..n	Vehicle Descriptor Element (17.8) [OV]							

Table 41: Fleet Management reply message

The fields *Target Application ID, ApplFlag1, ControlFlag1*, are exactly what was sent in the *Fleet Management Configuration Update Message #1*. If no *Fleet Management Configuration Update Commit Message* is expected, then a value of *OK* in the *error element* indicates the result of the Fleet Management Configuration Request. Sending a *Fleet Management Configuration Update Message* with *ApplFlag1* set to *no change* allows the SO to query about the provisioning status of a TCU.

16.2.1.4 Fleet Management Configuration Request Message

This message is sent from the TCU to the SO to request a Fleet Management Configuration update. It is typically used if a unit has been authorized for a service, and is either nearing end of that authorized period or has elapsed into a grace period. The format of the response is identical to *Fleet Management Configuration Reply* Message except the message id indicates a *Fleet Management Configuration Request M*essage.

If the *Fleet Management Configuration Request Message* is sent from SO to TCU, the SO is requesting a *Fleet Management Configuration Status M*essage. This allows the SO to determine the provisioning status of an application in a TCU.

Octet \ bit	0	1	2	3	4	5	6	7
1…3	Application Header (6)							
4…8	Version (17.4) [OV]							
9	More Flag	Target Application ID (4.1)						
10	ApplFlag1 (17.1)		ControlFlag1(17.2)					
11	StatusFlag1 (17.10)		Reserved, set to 0					
12-15	Start Time (17.5) [O]							
16-19	Expiration Time (17.11) [O]							
20	Error element (17.13) [O]							
21	Vehicle Descriptor Element (17.8) [OV]							

Table 42: Fleet Management status request

16.2.1.5 Fleet Management Configuration Status Message

This message is sent from the TCU to the SO in response to the SO *Fleet Management Configuration Request Message*. It is used to obtain the current Fleet Management configuration data for the SO in the event the SO does not have it or to verify that the configuration data is still valid. The format of the response is identical to *Fleet Management Configuration Update Message* except the message id indicates a *Fleet Management* Configuration *Status M*essage.

The *Fleet Management Configuration status* message is also used at the end of a configuration sequence so that the SO can verify the correct status.

16.2.1.6 Fleet Management Configuration Edit Message

This message is sent from the SO to the TCU. It is used to modify the current Fleet Management configuration. The format of the response is identical to *Fleet Management Configuration Update Message* except the message id indicates a *Fleet Management* Configuration *Edit M*essage. The modification may add a new data collection type, replace an existing data collection type, or delete a current data collection type by indicating a zero length field for the data collection type. The Fleet Management Configuration Reply message is used to indicate a successful modification.

17 Information Elements

17.1 ApplFlag1

This table defines the provisioning to perform. If *provision update commit message* is used, then this may differ between Provision update message #1 and Provision update commit message. The value used in *provision update commit message* shall define the action to be taken.

Value	Definition
0	No change to application
1	Activate application
2	Deactivate application
3	Change for this application

Table 43: ApplFlag1 values

17.2 ControlFlag1

This is defined as a 6 bit field with bit 0 on the left and bit 5 on the right. Remember that this field does not begin on a byte boundary.

Bit number	Definition
0	1=A 2nd byte ControlFlag2 exists 0=This is the last Control Flag
1	1=Grace Time is present 0=Grace Time is absent
2	1=Start Time field is present 0=Start Time field is absent
3	1=End Time or Expiration Time field is present 0=End Time or Expiration Time field is absent
4	1=Vehicle Descriptor element is present 0=Vehicle Descriptor element is absent
5	1=Provisioning uses *provision update commit message* 0=Provisioning does not use *provision update commit message*

Table 44: ControlFlag1 values

17.3 ControlFlag2

Bit Number	Definition
0	1=Another control Flag octet follows this octet. The next control Flag is not currently defined, but this definition allows for future expansion. 0=This is the last Control Flag
1-3	0=Use the sample method of collection. 1=Use the profile method of collection. 2-7 Reserved
4	1=Number of Samples is present 0=Number of Samples is not present

5	Reserved, set to 0
6	Reserved, set to 0
7	Reserved, set to 0

17.4 Version element

Octet \ Bit	0	1	2	3	4	5	6	7
1	IE Identifier		More flag	Length (number of octets that follow)				
2	Car Manufacturer ID							
3	TCU manufacturer ID							
4	Major hardware release							
5	Major software release							

Table 45: Version element encoding

17.4.1 IE Identifier

The value is 0 for type binary.

17.4.2 Length

This value is 0 if the information element is absent. Otherwise it specifies the number of octets that follow.

17.4.3 Car Manufacturer ID

This just identifies the vehicle in which the hardware/software product is resident. It is informational. This informative to the receiver of the message who should act upon this information if it is a version it can not handle.

Value	Definition
0	Mercedez-Benz
1	Renault
2	GM Opel
3	Porsche
4	Ford
5	PSA
6-127	Not currently used
128-255	reserved

Table 46: Manufacturer ID Defintions

17.4.4 TCU Manufacturer ID

This just identifies the source of the hardware/software product. It is informational. This informative to the receiver of the message who should act upon this information if it is a version it can not handle.

Value	Definition
0	reserved
1	Motorola
2	Nokia
3	Becker
4	Clarion
5-127	Not currently used
128-255	reserved

Table 47: TCU Manufacturer ID Definitions

17.4.5 Major hardware release

This identifies the current revision of the hardware. This informative to the receiver of the message who should act upon this information if it is a version it can not handle.

17.4.6 Major software release

This identifies the current revision of the software release. This informative to the receiver of the message who should act upon this information if it is a version it can not handle.

17.5 Start Time

This is the time at which service is to start. If it is absent, then it is the current time or another predefined time. The format is defined in section 17.29.

17.6 End time

This is the time at which service is to end. If it is absent, then it no end time is defined. This field is only required when activating a service and establishing the ending time. If a service is to be deactivated and this element is not present, then the deactivation is immediate. If this element is present for a service deactivation, then it specifies the date where the service ends. The format is defined in section 17.29.

17.7 Grace time

If supported in the TCU, this is the time after *end time* whereby service for the application is still supported. The format is defined in section 17.29.

17.8 TCU Descriptor

The intent of this information element is to provide version information about the TCU and the environment in which it operates when it is sent from TCU to SO. If this information element is in a message from SO to TCU, it specifies the latest version level at which the SO is capable of running. If no version information is present, then this element shall have a length of 0.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier		More flag	Length				
2	Reserved							
3	TCU Version element 1 (17.9)							
4..k-1	:							
k	TCU Version element n							

17.8.1 IE Identifier

The values are defined according to section 5.4. Currently this value is set to 0.

17.8.2 Length

This field defines the number of octets that follow in this information element

17.9 TCU Version

This element is intended to provide for more specific versioning information as needed for

- TCU
- Components in the TCU
- External interfaces to the TCU

The TCU can respond as part of provisioning with the version information.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier		length					
2	Device ID							
3	IE Identifier		length					
4	Version ID							

17.9.1 IE Identifier

The values are defined according to section 5.4.

17.9.2 Length

The length is defined as in section 5.4.

17.9.3 Device ID

This describes the device for which we want the version number. The device can be a number and taken from the following device table, or it can be a character string for devices not in the table. The currently defined devices are

Value	Definition
1	TCU hardware version number
2	TCU hardware manufacturer number
3	TCU software version number
4	TCU CAN version number
5	ACP transport layer version number
6	ACP application layer version number
7-255	

17.9.4 Version ID

This specifies the version information for the specified device. The version number can be a number and taken from the following version table, or it can be a character string for version not in the table. The currently defined decimal versions are

Value	Version
1	1
2	2
3	3
4	4
5	5
:	:
n-255	

If the binary version number is not adequate, then the version information may be encoded as a character string as defined by section 5.4.

17.10 Provisioning StatusFlag1

This indicates the current provisioning status of the application prior to reception of the *Provision Update message #1*.

Value	Definition
0	Application already provisioned
1	Application not already provisioned
2	See Error element for more details
3	Reserved

17.11 Provisioning TCU Response Flag

This indicates the reason for this reply message.

Value	Definition
0	Reserved
1	TCU initialize mode
2	TCU response to *provision update message #1*
3	TCU response to *provision commit message*

17.12 Expiration Time

If the application is already provisioned and has a provisioning expiration time, then that value is in this field. At this time, the specified application will stop working. The format is the standard Time Stamp format as defined in (17.29).

17.13 Error Element

The field *Error element* is a 1 octet field that can take on values in range 0-255. The format of this field follows the definition in (5.4). It is defined so that it can be replaced by textual definitions as needed in special circumstances. If the 1 octet is not sufficient, then a message of 2 octets or longer can be used.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier		length					
2..n	Error Octet or Error Message							

Table 48: Error Element

Decimal value	Description
0	Everything OK, no mistake
1	Service currently not available
2	Incorrect application
3	Unknown version
4	Unknown message type
5	Unknown data in message
6	Unknown transport version
7	Data error in transport frame
8	Security violation
9	No access; no customer
10	No access; service not available
11	Not access; authentication failure
12	No access; other reasons
13	Invalid session ID
14	Reserved
15	Language not supported
16	Descriptor in *Provision Update Message #1* does not match data in the TCU
17	SIM identifier in the TCU is not the provisioned SIM

18	Provisioning layer received, but unable to process
19	General non-specific error
20	No access to protected phonebook on SIM card
21	Unable to write to EEPROM
22	Invalid phone number
23	VIN does not match
24	Vehicle type does not match
25	Provisioning request not processed. Too many target applications specified. Provision with a smaller group.
26-255	Available

17.14 Location

This group defines the position history of the vehicle. The entries are ordered from most current to oldest.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier =2		More Flag=1	Length				
2	More Flag	Length						
3...n-1	Current GPSRawData (17.15) [OV]							
n...m-1	Prior GPSRawData (17.15) [OV]							
m...m+8	Current Dead Reckoning Data (17.22) [O]							
m+9...k-1	Array of Area Location Delta Coding (17.21) [V]							

Table 49: Location

The current GPSRawData and prior GPSRawData are only needed when the GPS is co-located with the TCU and, therefore, is optional.

17.15 GPSRawData

The GPS raw data for the first location is available for inverse DGPS.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier = 0		More flag	Length				
1..12	Area Location Coding (17.16)							
13	Number of Satellites				Reserved			
14	Satellite ID 1							
15…n	Satellite ID n							

Table 50: GPS Raw Data

The Satellite ID specifies the visible satellite when the measurement was taken. Each satellite id occupies 8 bits. A minimum of 3 is required for position determination. The more satellites visible, the more accurate a position determination possible.

17.16 Area Location Coding

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier = 0		More flag	Length				
2	More Flag	AreaLocationStatusFlag1 (17.19)						
3	More Flag	AreaLocationStatusFlag2 (17.20) [O]						
4	Area Type (Table 53:)			Location Type Coding (Table 54:)			Reserved	
5	More Flag	Time Difference						
6 - 9	Longitude							
10..13	Latitude							
14..15	Altitude							
16	Position Uncertainty Estimate (Table 52:)							Reserved
17	Heading Uncertainty Estimate (Table 55:)			Heading (3.1)				
18	Reserved set to 0					Distance Flag (15.5.9)	Time Flag (17.18)	
19	Velocity (17.17)							

Table 51: Area Location Coding

We provide only location information and allow the service operator to interpret these values for velocity, etc. information. This position is determined using GPS, but there exist times when GPS is unavailable. GPS position information can be augmented through other means such as Dead Reckoning. In the event GPS can not be augmented through other means, no position history will be available for that time period. When a valid position again becomes available, then the position history will again begin, but there will appear an atypical large time difference in the position history as well as typical geographical jump in position.

The time difference is the time that has elapsed between a prior GPS reading and the current GPS reading or the time difference between the current GPS reading and the actual cause for transmission.

Note that the fields in this element are fixed and may be optional even though they so indicated. The Length field has special use in that it can be less in 21 to indicate that data is not present. For example, a length of 13 indicates that Altitude and the remaining fields are not present.

The heading are multiples of 15 degrees. For example, a heading value of 3 is 45 degrees or North East.

The values of *area type* are defined in Table 53:

The units for time difference is 1 second with a maximum value of 127 seconds. It specifies the time since the prior location entry. If there is no prior location entry, then the entry is 0. If the time difference to the prior entry exceeds 127 seconds, then the more flag is used to indicate another octet of time difference follows the current octet. This can be used to determine the time between location measurements in a position history.

The latitude may vary from -90° to +90°. The number 0° corresponds to the equator. The units are in milliarcseconds and allow for the range from –324,000,000 to +324,000,000. One unit corresponds to approximately .03 meter. Positive values are North and negative values are South.

The longitude may vary from -180° to +180°. The number 0° corresponds to the Greenwich meridian. The value may range from –648,000,000 to +648,000,000. The units are in millisarceconds and allow for the range from –648,000,000 to +648,000,000. One unit corresponds to approximately .03 meter. Positive values are East of the Greenwich median and negative values are West.

The altitude may vary from –1000m to 18,000 meters. The number 0 corresponds to sea level. One unit corresponds to 1 meter.

The *position uncertainty estimate* is defined by a variation on the Binomial expansion. The uncertainty, expressed in meters, is mapped to a number K, with the following formula from [GSM332]:

$$R = C((1+x)^K-1)$$

Where C = 10 and x = 0.1. With 0 ≤ K ≤ 127, a suitable useful range between 0 and 1800 kilometers is achieved for the uncertainty, while still being able to code down to values as small as 1 meter. The uncertainty can then be coded as in table

Value of K	Uncertainty
0	0 m
1	1 m
2	2.1 m
--	--
20	57.3 m
--	--
40	443 m
--	--
60	3 km
--	--
80	20 km
--	--
100	138 km
--	--
120-	927 km
--	--
127	1800 km

Table 52: Position Uncertainty

The currently defined area definitions are defined in the following table. Currently only the point definition is included and is intended for TCUs that have a GPS unit for location determination. The point location combined with the *position uncertainty estimate* can be used to locate the TCU. If the TCU can only be located to a cell within the cellular network, then the *point* defines the center of the cell and the *position uncertainty estimate* is the approximate size of the cell.

Value	Definition
0	Point (latitude, longitude, altitude) in milliarcseconds increments
1	Point (latitude, longitude, altitude) in 100 milliarcseconds increments
2-7	Reserved

Table 53: Area Type Definitions

The element *location type coding* defines the coordinate system used to define the location of the TCU. Location type coding is defined according to the following table:

Value	Definition
0	WGS 84
1-7	Reserved

Table 54: Location Type Coding

The *heading* is defined as the azimuth. The *heading uncertainty estimate* is defined in the following table.

Value	Heading uncertainty	Value	Heading uncertainty
0	≤ 6°	8	≤ 40°
1	≤ 9°	9	≤ 50°
2	≤ 12°	10	≤ 60°
3	≤ 15°	11	≤ 80°
4	≤ 20°	12	≤ 100°
5	≤ 25°	13	≤ 125°
6	≤ 30°	14	≤ 150°
7	≤ 35°	15	> 150°

Table 55: H eading Uncertainty estimate

17.17 Velocity

The velocity is expressed in distance/time where distance is defined as in section 15.5.9 and time is defined as in section 17.18. The normal definition is kilometers/hour. The field size allows for up to 254 kilometers/hour. The value 255 is reserved for velocities of 255 and larger.

17.18 Time Flag

Value	Definition
0	Seconds
1	Minutes
2	Hours
3	Reserved

Table 56: TimeFlag encoding

17.19 AreaLocationStatusFlag1

Bit	Definition
1	Reserved and set to 0
2	1=No 3D fix available 0=Is using 3D fix
3	1=No 2D fix available 0=Is using 2D fix
4	1=Position data is not valid 0=Position data is valid
5	1=Differential GPS is being used 0=Differential GPS not being used
6	1=Heading data is not valid 0=Heading data is valid
7	1=Almanac is bad 0=Almanac is good

Table 57: AreaLocationStatusFlag1

17.20 AreaLocationStatusFlag2

Bit	Definition
1	Reserved
2	0=Old GPS data used from EPROM
	1=New GPS data from satellites
3	Reserved, set to 0
4	Reserved, set to 0
5,6,7	0=North 1=North East 2=East 3=South East 4=South 5=South West 6=West 7=North West

Table 58: AreaLocationStatusFlag2

17.21 Area Location Delta Coding

This element defines a more compact form for specifying the position history of a vehicle.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier = 0		More flag	Length				
2	More Flag	Delta Longitude 1						
3	More Flag	Delta Latitude 1						
		: :						
4		Delta Longitude n						
n		Delta Latitude n						

Table 59: Area Location Delta Coding

This 1 octet fields allow for a value range of –64 to +63 milliarcseconds or 100 milliarcseconds increments as defined by Area Type in section 17.16. This provides for 3.1 meter resolution which is an acceptable compromise for standard maps having 5 meter resolution. This also reduces the computational requirements of the TCU. Since the range for milliarcseconds is ±63 (100s) milliarcseconds, this equates to a distance of ±195 meters at the equator and gets smaller as the north/south pole is approached.

17.22 Dead Reckoning Data

The Dead Reckoning Data follows the WGS4 format. The specific format for Dead Reckoning is extracted from the WGS84 reference ellipsoid. It is the current position including the dead reckoning data.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier = 0		More flag	Length				
2...5	Latitude							
6...9	Longitude							

Table 60: Dead Reckoning Data

17.23 Vehicle Descriptor

The intent of this information element is to define the vehicle to the service operator. If the vehicle is already known to the service operator through some prior arrangement, then the only field required here is the VIN number. If the vehicle is not previously known to the service operator, then all fields are required. This is necessary for accounting reasons.

If this element group is in a configuration message then the following interpretations hold. If the length part of the element is 0, then the element is absent. This means no change from its previous value. To initially establish a value for an element requires a non-zero element.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier = 0		More flag	Length				
2	Vehicle Descriptor Flag1 (Table 62:)							
3	Vehicle Descriptor Flag2 (Table 63:) [O]							
4	Language (17.23.2) [O]							
5	Model year (17.23.9) [O]							
6..p-1	VIN (17.23.4) [OV]							
p..n-1	TCU serial # (17.23.5) [OV]							
n..m-1	License Plate (17.23.6) [OV]							
m..k-1	Vehicle Color (Table 67:) [OV]							
k..j-1	Vehicle Model (Table 68:) [OV]							
j…I-1	IMEI (17.23.9) [OV]							

Table 61: Vehicle Descriptor

17.23.1 Vehicle Descriptor Flag

In Table 62: a bit value of 0 indicates the element is absent while a bit value of 1 indicates the element is present. If the bit indicates the element is present, but the length of its element is 0, then the field has no data following it. If the length is not 0, then the element specifies the new value. If all bits are 0, then no elements are present, and only 1 octet of storage is used.

Octet \ bit	0	1	2	3	4	5	6	7
1	More Flag	Language	VIN	TCU serial #	Vehicle Color	Vehicle Model	License Plate	IMEI

Table 62: Vehicle and Cargo Descriptor Flag 1

Octet \ bit	0	1	2	3	4	5	6	7
1	More Flag	Model year	0	0	0	0	0	0

Table 63: Vehicle Descriptor Flag 2

17.23.2 Language

These are just the languages as defined by ISO 639. This element is one octet long. This defines the recommended language used during voice communications. It also indirectly defines the recommended character set for text strings. A copy of ISO 639 is in Table 64:

Language	Assigned Number	Language	Assigned Number
Afar	0x01	Maori	0x47
Abkhazian	0x02	Macedonian	0x48

Language	Assigned Number	Language	Assigned Number
Afrikaans	0x03	Malayalam	0x49
Amharic	0x04	Mongolian	0x4A
Arabic	0x05	Moldavian	0x4B
Assamese	0x06	Marathi	0x4C
Aymara	0x07	Malay	0x4D
Azerbaijani	0x08	Maltese	0x4E
Bashkir	0x09	Burmese	0x4F
Byelorussian	0x0A	Nauru	0x81
Bulgarian	0x0B	Nepali	0x51
Bihari	0x0C	Dutch	0x52
Bislama	0x0D	Norwegian	0x53
Bengali;	0x0E	Occitan	0x54
Tibetan	0x0F	(Afan)Oromo	0x55
Breton	0x10	Oriya	0x56
Catalan	0x11	Punjabi	0x57
Corsican	0x12	Polish	0x58
Czech	0x13	Pashto	0x59
Welsh	0x14	Portuguese	0x5A
Danish	0x15	Quechua	0x5B
German	0x16	Rhaeto-Romance	0x8C
Bhutani	0x17	Kirundi	0x5D
Greek	0x18	Romanian	0x5E
English	0x19	Russian	0x5F
Esperanto	0x1A	Kinyarwanda	0x60
Spanish	0x1B	Sanskrit	0x61
Estonian	0x1C	Sindhi	0x62
Basque	0x1D	Sangho	0x63
Persian	0x1E	Serbo-Croatian	0x64
Finnish	0x1F	Sinhalese	0x65
Fiji	0x20	Slovak	0x66
Faeroese	0x82	Slovenian	0x67
French	0x22	Samoan	0x68
Frisian	0x83	Shona	0x69
Irish	0x24	Somali	0x6A
Scots-Gaelic	0x25	Albanian	0x6B

Language	Assigned Number	Language	Assigned Number
Galician	0x26	Serbian	0x6C
Guarani	0x27	Siswati	0x6D
Gujarati	0x28	Sesotho	0x6E
Hausa	0x29	Sundanese	0x6F
Hebrew	0x2A	Swedish	0x70
Hindi	0x2B	Swahili	0x71
Croatian	0x2C	Tamil	0x72
Hungarian	0x2D	Telugu	0x73
Armenian	0x2E	Tajik	0x74
Interlingua	0x84	Thai	0x75
Indonesian	0x30	Tigrinya	0x76
Interlingue	0x86	Turkmen	0x77
Inupiak	0x87	Tagalog	0x78
Icelandic	0x33	Setswana	0x79
Italian	0x34	Tonga	0x7A
Inuktitut	0x89	Turkish	0x7B
Japanese	0x36	Tsonga	0x7C
Javanese	0x37	Tatar	0x7D
Georgian	0x38	Twi	0x7E
Kazakh	0x39	Uighur	0x7F
Greenlandic	0x8A	Ukrainian	0x50
Cambodian	0x3B	Urdu	0x21
Kannada	0x3C	Uzbek	0x23
Korean	0x3D	Vietnamese	0x2F
Kashmiri	0x3E	Volapuk	0x85
Kurdish	0x3F	Wolof	0x31
Kirghiz	0x40	Xhosa	0x32
Latin	0x8B	Yiddish	0x88
Lingala	0x42	Yoruba	0x35
Laothian	0x43	Zhuang	0x3A
Lithuanian	0x44	Chinese	0x41
Latvian,	0x45	Zulu	0x5C
Malagasy	0x46		

Table 64: ISO 639 language encoding

17.23.3 Model Year

This is the model year for the vehicle. A 2 digit year is sufficient.

17.23.4 VIN

By default this is the VIN number. The only requirement on this field is an identifier that uniquely identifies the wireless device consumer. For element format see section 5.4

Octet \ bit	0	1	2	3	4	5	6	7
1	IE identifier = 1		More flag	Length				
2..n-1	Text string							

Table 65: VIN

17.23.5 TCU Serial

This is the TCU Serial #. The only requirement on this field is an identifier that uniquely identifies the wireless device consumer. For element format see section 5.4

Octet \ bit	0	1	2	3	4	5	6	7
1	IE identifier = 1		More flag	Length				
2..n-1	Text string							

Table 66: TCU Serial #

17.23.6 License Plate

This is the vehicle license plate number.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE identifier = 1		More flag	Length				
2..n-1	Text String							

Table 67: License Plate

17.23.7 Vehicle Color

This is a text string defining the vehicle color.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE identifier = 1		More flag	Length				
2..n-1	Text String							

Table 68: Vehicle Color

17.23.8 Vehicle Model

This is the human readable form identifying the make and model of the vehicle.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE identifier = 1		More flag	Length				
2..n	Text String							

Table 69: Vehicle model

17.23.9 IMEI

This number is the equipment serial number of the mobile equipment providing wireless equipment. In GSM this is the International Mobile Equipment Identifier.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE identifier = 2		More flag	Length				
2..n	BCD							

Table 70: Vehicle model

17.24 Breakdown Status

This group defines the nature of the breakdown.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier = 0		More flag	Length				
2	Breakdown Source (17.25)							
3	Breakdown Sensor (17.26)							
4…n-1	Breakdown Data (17.27) [OV]							

Table 71: Breakdown Status

17.25 Breakdown Source

This 1 octet field defines the cause for the breakdown. The bit *Tow truck needed* defines those lesser emergency cases where only a vehicle service person is necessary. The urgency of the response is less for *Tow truck needed*.

Bit	Breakdown Definition
0	1=then another octet of breakdown source follows this octet 0=this is the last octet of breakdown source
1	Manually activated
2	Vehicle rolled
3	Air bag activated
4	Crash sensor activated
5	Floating car data input
6	Tow truck needed
7	Vehicle initiated theft tracking

Table 72: Breakdown Source

Bit	Breakdown Definition
0	1=another octet of breakdown source follows this octet 0=this is the last octet of breakdown source
1	Vehicle is started (ignition on)
2	Vehicle is turned off (ignition off)
3	Vehicle is moved
4	Other sensor activated
5	Re-send location (TCU button pressed)
6	Re-send location (SO sent message)
7	Reserved

Table 73: Breakdown Source extended octet

17.26 Breakdown Sensors

For automatically detected emergencies, this byte defines the sensor within the class *Breakdown source* that initiated the emergency request.

Bit	Crash sensors definition
0	More flag
1	Rollover sensor activated
2	Front sensor
3	Rear sensor
4	Side sensor
5	Vehicle Alarm activated
6	Reserved
7	Reserved

Table 74: Breakdown Sensors

17.27 Breakdown Data

This allows for additional descriptive data to be included with the emergency notification message.

Octet \ bit	0	1	2	3	4	5	6	7
1	Data type (Table 76:)		More flag	Length				
1-n	Breakdown Data							

Table 75: Breakdown Data

The following table defines the values in data type field in element *Breakdown data*.

Value	Breakdown Reason Type
0	Breakdown Code
1	User defined binary data
2	Text
3	Crash Data

Table 76: Breakdown Data Type Values

17.28 Cargo

This is a text item defined as in section 5.4. It describes what this vehicle contains. It is primarily intended for commercial vehicles where the vehicle and/or cargo may require special treatment in an accident or other types of emergencies. If length is 0, then this field is absent. The following information element allows the cargo to be described containing the same information as in earlier ACP versions.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE identifier = 0		More flag	Length				
2..n	Transport Good Number (see Table 78:) [V]							

Table 77: Cargo (Tranport good number encoding)

The number of *Transport Good Number* entries can be determined by the *length* field.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE identifier = 0		More Flag	Length				
2-3	Coded number							
4-5	Quantity							
6	Units							

Table 78: Transport Good Number entry

17.28.1 Coded number

This is suggested to be the Hommel transport good number which is a 4 digit decimal number.

17.28.2 Quantity

This can range from 0 to 65,535.

17.28.3 Units

The units for the field *quantity*.

The following information element allows the cargo to be described as a text string.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE identifier = 1		More flag	Length				
2..n	ASCII Text String [V]							

Table 79: Cargo (ASCII encoding)

17.29 Time Stamp

The Time Stamp is always encoded as UTC time. It is the time at which the triggering event in the TCU occurred.

Octet \ bit	0	1	2	3	4	5	6	7
1	Year(00=1990, 01=1991 ranges up to 62=2052)						Month of year range (1…12)	
2	Month of year		Day of month, range (1…31)					Hour of day
3	Hour of day, range (0…23)				Minutes, range (0…59)			
4	Minutes		Seconds, range (0…59)					

Table 80: UTCTime

17.29.1 Year

This specifies the year beginning in 1990. The field allows for a year of up to 2052. The value of 63 is reserved for future expansion if needed.

17.29.2 Month

This field represents the month of the year which ranges from 1 to 12 where 1 is January and 12 is December.

17.29.3 Day

This field represents the day of the month and ranges from 1 to 31.

17.29.4 Hour

This field represents the hour of the day and ranges 0 to 23.

17.29.5 Minute

This field represents the minute of the hour and ranges from 0 to 59.

17.29.6 Seconds

This field represents the seconds of the minute and ranges from 0 to 59.

17.30 Serial Number

This element represents the serial number of the TCU. If the value of *length* is 0 then the field is absent.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier = 1		More flag	Length				
2…n	ASCII text string							

Table 81: Serial Number

17.31 Confirmation

This element represents the confirmation status of an assistance reply message.

Bit	Confirmation Definitions
0	1=another octet follows this one with additional control information 0=If 0, then this is the last octet of control information.
1	1=Assistance notification accepted and processing continues 0=Assistance notification rejected, call 112 or 911 directly
2	1=Turn Speaker On 0=Turn Speaker Off
3	1=Processing [tracking] should start or continue 0=Processing [tracking] should stop

Table 82: Confirmation

17.32 Ecall Control Flag 2

This element defines additional control functions in the ecall message.

Bit	Confirmation Definitions
0	1=Another octet follows this one with additional control information 0=This is the last octet of control information
1	1=Cancel the alarm 0=No alarm related action
2	1=A group follows the last control flag (More Flag, Transmit Interval, Phone Number 0=No group follows the last control flag
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved

Table 83: Ecall Control Flag 2

17.33 Transmit Units

The value in this element is only valid if the *Confirmation* element indicates tracking should start or continue. The values are interpreted as in the following table. It specifies the interval at which an emergency call message is sent on a regular basis. The purpose is to track a vehicle.

Value	Transmit Units
0	Second
1	Minute
2	Hour
3	Send emergency call message one more time
4	Send only one message

Table 84: Transmit Units

17.34 Phone Number

This provides for a variable number of Phone Number elements. Each Phone Number element is variable in length. If multiple phone number elements appear for a given application, it is up to the application to determine when and how to use them serially or concurrently. The level of concurrency may be dictated by norms or legal issues within a country. In general the order that *Phone Number* elements appear specifies the order that the application shall attempt to use the specified bearers.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier = 0		More flag	Length				
2	Control Flag Present (17.34.2	Numbering plan (17.34.5)		Reserved	Bearer (17.34.6)			
3	Control Flag (17.34.3) [O]					Phone Mode [O]		
4-6	Country code (Binary) [O]							
7-8	Network code (Binary) [O]							
9..n	Digits (17.34.7) [V]							

Table 85: PhoneNumber

17.34.1 Length

The length is the number of octets that follow. A length of 0 means the element contains no data. The maximum value is 24 bytes or 48 digits.

17.34.2 Control Flag Present

If set, then the field *Control Flag* byte is present and the *Phone Mode* byte is present.

17.34.3 Control Flag

This byte specifies additional fields that are present.

Bit	Definition
0	1=A 2nd control flag follows this control flag 0=This is the last control flag in this element
1	1=Mobile *Country Code* byte is present 0=Mobile *Country Code* byte is not present
2	1=Mobile *Network Code* byte is present 0=Mobile *Network Code* byte is not present
3	Reserved set to 0
4	Reserved, set to 0

Table 86: ControlFlag encoding

17.34.4 Phone Mode

This field only exists when a phone supports multiple mobile networks as in bi-mode and tri-mode phones. If the *bearer* defined can be from multiple supported networks, then the non-zero value defines the network to initially use.

Value	Definition
0	Reserved not used
1	Use default network of phone
2	Use 2nd choice
3	Use 3rd choice
4-7	Reserved not used

Table 87: PhoneMode encoding

17.34.5 Numbering Plan

This field defines the numbering plan type. National numbers only work within the boundaries of a single country. International numbers work regardless of country in which the vehicle is located. The service operator decides how to support their subscriber according to their business plans and any other business issues.

Value	Definition
0	Numbering plan is unknown
1	National number
2	International number (begins with country code)
3 - 7	Reserved

Table 88: Numbering plan definitions

17.34.6 Bearer

The currently supported bearers are

- Voice
- CDPD
- CSD (CDMA, GSM, iDEN, IS-136, PDC, or PHS)
- FLEX
- Packet Data (CDMA, GSM, iDEN, IS-136, or PDC)
- R-Data
- ReFLEX
- SMS (CDMA, DataTAC, GSM, or IS-136)
- USSD.

These are intended as generic names to be mapped on appropriate bearer of the underlying wireless network. For instance for GSM, short message service maps onto SMS-PP.

Value	Definition
0	No bearer specified
1	SMS (destination address)

Value	Definition
2	SMSC (store and forward address or relay address)
3	ISDN Numbering Plan (Voice domestic or roaming)
4	ISDN Numbering Plan (112 voice)
5	ISDN Numbering Plan (Data)
6	GPRS
7	CSD
8	FLEX
9	Packet Data
10	R-Data
11	USSD
12	CDPD
13	ISDN numbering plan voice roaming
14-15	Reserved

Table 89: Bearer definitions

17.34.7 Digits

This defines the phone number as specified in the *digits* element. The digits are in packed decimal format. The number implies a bearer as specified in *bearer* element. If the number of *digits* is odd, then the element is terminated with a packed decimal digit of F. The special characters of *, #, and + are encoded as hex characters from the range 0xA to 0xE and are defined in Table 90: .

Special character	Bit representation
*	0xA
#	0xB
+	0xE

Table 90: Special characters in phone number

17.35 Control Function

This element represents the remote vehicle device to control. If the value of *length* is 0 then the field is absent.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier = 0		More flag	Length				
2	Control Device							
3	Reserved				Transmit Units (17.33) [O]			
4	Transmit Interval [O]							

Table 91: Control Function

Value	Control Function Definition
0	Door Locks
1	Start Vehicle Tracking
2	Covert Mode
3	Mute the microphone
4	UnMute the microphone
5	Transmit Interval
6	Stop Vehicle Tracking
7	Vehicle Tracking with commit
8	Commit to Vehicle Tracking
9-255	Reserved

17.36 Function Command

This element represents the action required of the remote vehicle device If the value of *length* is 0 then the field is absent.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier = 0		More Flag	Length				
2	Function Command							
3…n-1	Action Parameter Variable							

Table 92: Function Command

Value	Function Command Definition
0x00	Permit
0x01	Reject
0x02	Enable
0x03	Disable
0x04…0xFF	Reserved

Action Parameter Variable	Action Parameter Variable Definition
0…255	TBD

17.37 Function Status

This element represents the status of the remote vehicle device after the remote vehicle function has been performed. If the value of *length* is 0 then the field is absent.

Octet \	0	1	2	3	4	5	6	7

bit								
1	IE Identifier = 0		More flag	Length				
2	Device Status							

Table 93: Device Status

Value	Definition
0x00	Doors Are Locked
0x01	Doors Are Unlocked
0x02…0xFF	Reserved

17.38 Information Type

The following table details the various types if information that the end user may request.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier = 0		More flag	Length				
2	More Flag	Information Type						
3..n	Information (17.39)[O]							

Table 94: Information Type

Information Requested Value	Information Requested Definition
0	Reserved
1	Verbal Information
2	Stock Information
3	Travel Route Information
4	Hotel Information
5	Traffic information requested (verbal)
6	Traffic information requested (automated)
7	ASCII Text String
8	Point of Interest
9	Cargo
10	Private
11…127	Reserved

Table 95: Information Requested

17.39 Raw Data

This element may be used for any raw data that may need to be transmitted between the TCU and SO.

Octet \ bit	0	1	2	3	4	5	6	7
1	IE Identifier = 0		More flag	Length				
2…n-1	Raw Data							

Table 96: Raw Data

17.40 Diagnostic Data Service

This is an identifier that signifies what type of diagnostic service is being requested or reported. The SO sends the request for information and the TCU reports the information.

Value	Diagnostic Data Service Description
0	Enter Diagnostic Mode
1	Exit Diagnostic Mode
2	Diagnostic Failure Code
3	All Diagnostic Failure Codes
4	Parameter Data
5	Control Command
6	Diagnostic Self test
7	Diagnostic Failure Codes
8	Diagnostic Self test results
9-127	Reserved
128-255	Reserved for private use

Table 97: Diagnostic Data Service

17.41 Diagnostic Data Source

This is the module that the diagnostic data is originating from. Diagnostic data may originate from the TCU or from other modules on the vehicle bus. The table currently contains possible modules on the vehicle bus but is dependent upon the vehicle manufacturer and the vehicle model.

Value	Diagnostic Data Source Description
0	TCU
1	Anti-lock Breaking System (ABS) module
2	Power train Control Module (PCM)
3	Driver’s Door Module (DDM)
4	Driver’s Seat Module (DSM)
5	Lighting Control Module (LCM)
6	Engine Control Unit (ECU)
7	(ESP)
8-127	Reserved
128-255	Reserved for private use

Table 98: Diagnostic Data Source

17.42 Diagnostic Data

This is the actual diagnostic data that is being sent by the TCU or requested by the SO.

Octet \ bit	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	IE identifier = 0		More flag	Length				
2..n-1	Binary Data							

Table 99: Diagnostic Data

18 Appendix A: ACP 2.2 Authorization For Backward Compatibility

18.1 Authorization Messages

18.2 Initiate Service Authorization Message (SO -> CU)

This message is used by the SO to trigger the authorization sequence from the CU. It can be used when first providing ECall authorization to a CU. It can also be used at any other time for the SO to initiate a sequence whereby the CU is denied ECall authorization.

18.2.1 General Description

- Telephone number for mobile unit to call to begin authorization.
- SO test number used to check out if it works
- Service code number
- voice phone number for emergency voice phone call
- VIN
- Test calling number of SMSC
- Trigger for authorization acknowledgment request

All phone numbers are 4 bit packed decimal. In order to recognize the end of the phone number that is less than 16 digits in the 8 byte field, the last digit is followed by 4 bit packed decimal value of all 1's.

18.2.2 Message Type

Message type is ISAMsg. Authorization MessageType field is 0x04.

18.2.3 Message Structure and Content

Field	Type	bytes	Description
AuthorizationBitFlags	uint8	1	0x01 = 0 This authorization is not for SMS = 1 This authorization is for SMS 0x02 = 0 This authorization is not for BS = 1 This authorization is for BS 0x04 = 1 This authorization is for ECALL = 0 This authorization is not for ECALL 0x08 = 1 This authorization is for taxi ECALL = 0 This authorization is not for taxi ECALL 0xF0 Reserved; currently set to 0
Reserved	uint8	8	Reserved
SMSNumber	uint8	8	The ECN phone number of the SO to be used for authorization and ECALL messages using SMS.
BS24AuthorizationNumber	uint8	8	The ECN phone number of the SO to be used for authorization using BS24.
BS24Number	uint8	8	The ECN phone number of the SO to be used for BS24 taxi & emergency calls.

BS26Number	uint8	8	The ECN phone number of the SO to be used for BS26 calls.
SMSCNumber	uint8	8	The phone number of the SMSC to be used for SMS calls.
GPRSNumber	uint8	8	The ECN phone number of the SO to be used for GPRS calls. Currently this field is set to all 0.
SOBSTestNumber	uint8	8	The test phone number of the service operator to be used in ECall message test call.
Reserved	uint8	8	Reserved
CUSMSPhoneNumber	uint8	8	The phone number of the TCU. This number appears in the **callersNumber** field of the ACP header when packets are sent from the CU to the SO using SMS.
CUBSPhoneNumber	uint8	8	The phone number of the TCU This number appears in the **callersNumber** field of the ACP header when packets are sent from the CU to the SO using BS.
Reserved	uint8	8	Reserved
VINNumber	unit8	17	The unique Vehicle Identification Number. This consists of ascii characters.
VehicleTypeID	uint8	1	4 bit coded decimal established by the service operator. The range is 00 to 99. This field records the vehicle type, such as taxi, van, truck, bus, etc. 0x02 passenger vehicle 0x12 taxi
total message bytes		115 bytes	

18.2.4 Triggering Event

The service operator has received the emergency service subscription form the owner. The service operator then sends this message to the CU being authorized. The intent is to trigger the unit into an authorization check sequence, which is used for both start of service and service denial.

18.2.5 Responses

See the Service Authorization Check Message.

18.2.6 System Issues

The CU must respond with an Authorization Check Message within a specified time in order for the authorization sequence to continue.

All phone numbers are 4 bit packed decimal. In order to recognize the end of the phone number in the 8 byte field, the last digit is followed by 4 bit packed decimal value of all 1's.

18.3 Service Authorization Check Message (CU -> SO)

18.3.1 General Description

The caller's number used by the CU can be retrieved from the packet header.

- Service Code
- Request for authorization

18.3.2 Message Type

Message type is SACMsg. Authorization MessageType field is 0x05.

18.3.3 Message Structure and Content

Field	Type	bytes	Description
Authorization bit flags	uint8	1	0x01 = 0 This authorization is not for SMS = 1 This authorization is for SMS 0x02 = 0 This authorization is not for BS = 1 This authorization is for BS 0x04 = 1 This authorization is for ECALL = 0 This authorization is not for ECALL 0x08 = 1 This authorization is for taxi ECALL = 0 This authorization is not for taxi ECALL 0xF0 Reserved; currently set to 0
Confirmation	uint8	1	0x80 = acceptance, authorization sequence continuing 0x01 = VIN does not match 0x02 = Message mismatch SO will have to retry 0x04 = Vehicle type does not match 0x40 = Authorization Sequence Complete
AuthorizationStatus	uint8	1	0x01 = 0 not authorized for SMS = 1 authorized for SMS 0x02 = 0 not authorized for BS = 1 authorized for BS 0x04 = 1 authorized for ECALL 0x08 = 1 authorized for taxi ECALL 0xF0 Reserved; currently set to 0
AuthorizationDate	uint8	4	Bit 0 & Bit 1 0 1 TCU initialize mode 1 0 TCU response to ISA 0 1 TCU response to SAR 1 1 Reserved bit 2-5: currently not used, set to 0

			bit 6-10: day of month, range (1..31) bit 11-14: month of year, range (1..12) bit 15-20: year (00 = 1990 01 = 1991 ranges up to 63 = 2053) bit 21-25: hours, range (0..23) bit 26-31: minutes, range (0..59)
Total message bytes		7 bytes	

18.3.4 Triggering Event

This message is sent when authorization is about to expire or an Initiate Service Message arrived from the SO.

This message is sent by the CU. It can be used as response to an ISA or SAR during initial authoirzation or reauthorization of services. When the ECall service is about to expire, the CU will attempt to connect to the SO using this message to initiate the reauthorization sequence. The authorizationDate field will contain the date of the last successfull [de]authorization sequence. It will only be zero for a CU that has never been authorized or is not currently authorized.

The attempt to call shall occur after the CU has been initialized (i.e. the ignition is turned on) and the CU has camped onto the GSM network. The attempts to reauthorize shall cease when the grace period expires. The ECall message service is no longer authorized for the vehicle after the grace period ends.

A field has been added to the SAC message for confirmation of authorization status. The authorization status field will be set as a confirmation when a second SAC has been sent in response to the SAR message. The SO will be sending an AP_ACK of the data in the second SAC

18.3.5 Response

See Service Authorization Response Message

18.3.6 System Issue

The CU will send an application negative response if its VIN disagrees with the message VIN from the ISAMsg. This is just the ACP positive acknowledgement with a piggybacked application message. This is like the CU saying, "I saw the SO message, but it does not make sense for me". This message is recognized by the confirmation field value of 0x10 in the message.

18.4 Service Authorization Response Message (SO -> CU)

18.4.1 General Description

The calling phone number used by the service operator can be retrieved from the packet header.

- Telephone number for mobile unit to call to begin authorization.
- SO test number used to check out if it works
- Service code number
- voice phone number for emergency voice phone call
- VIN
- Test calling number of SMSC
- Service authorization acknowledgment or denial

18.4.2 Message Type

Message type is SARMsg. Authorization MessageType field is 0x07.

18.4.3 Message Structure and Content

Field	Type	bytes	Description
AuthorizationBit Flags	uint8	1	0x01 = 0 This authorization is not for SMS = 1 This authorization is for SMS 0x02 = 0 This authorization is not for BS = 1 This authorization is for BS 0x04 = 1 This authorization is for ECALL 0x08 = 1 This authroization is for taxi ECALL 0xF0 Reserved; currently set to 0
Reserved	uint8	8	Reserved
SMSNumber	uint8	8	The ECN phone number of the SO to be used for authorization and ECALL messages using SMS.
BS24AuthorizationN umber	uint8	8	The ECN phone number of the SO to be used for authorization using BS24.
BS24Number	uint8	8	The ECN phone number of the SO to be used for BS24 calls.
BS26Number	uint8	8	The ECN phone number of the SO to be used for BS26 calls.
SMSCNumber	uint8	8	The phone number of the SMSC to be used for SMS calls.
GPRSNumber	uint8	8	The ECN phone number of the SO to be used for GPRS calls. Currently this field is set to all 0.
SOBSTestNumber	uint8	8	The test phone number of the service operator to be used in ECall message test call.
Reserved	uint8	8	Reserved
CUSMSPhoneNumber	uint8	8	The phone number of the TCU. This number appears in the **callersNumber** field of the ACP header when packets are sent from the CU to the SO using SMS.
CUBSPhoneNumber	uint8	8	The phone number of the TCU. This number appears in the **callersNumber** field of the ACP header when packets are sent from the CU to the SO using BS.
Reserved	uint8	8	Reserved
VINNumber	unit8	17	The unique Vehicle Identification Number. This consists of ascii characters.
VehicleTypeID	uint8	1	defined the same as vehicleTypeID in the Emergency message. set to 0x12 for taxi ecall
Confirmation	uint8	1	0x00 = deny service to this CU 0x80 = begin / renew service to this CU
total message bytes		116 bytes	

18.4.4 Triggering Event

The CU has verified that the calling line number matches the stored number of the service operator and The CU has sent a Service Activation Request Confirmation message.

18.4.5 Responses

Their can be no response to this message or a SAC response message. Only the ACP acknowledgement message shall be sent.

18.4.6 System Issues

The phone number fields are valid (non zero) only if confirmation field indicates beginning of renewing of service. If the ECall service not previously in effect, then the authorization period begins with the day on which this message is received by the CU. This could occur if the customer is a first time customer, or the SO had previously sent a service denial message to this CU.

If the service is currently in effect or within the grace period, then this message means that service is again renewed for a period beginning with the expiration date of the ECall service.

When authorization occurs via the BS service, the SO cannot verify the phone number of the CU since CLIP may not be available.

The CU compares the field contents of the SARMsg with the ISAMsg. If the two messages are different, then the CU resends the SACMsg until the CU receives 2 identical SARMsg's or n trials have passed.

All phone numbers are 4 bit packed decimal. In order to recognize the end of the phone number in the 8 byte field, the last digit is followed by 4 bit packed decimal value of all 1's.

19 Appendix B: ACP 2.2 ECall For Backward Compatibility

19.1 Emergency Related Messages

19.2 Emergency Call Message (TCU -> SO)

19.2.1 General Description

The emergency call message is generated in the TCU for transmission to a service operator. This message is sent via SMS or BS, triggered by an emergency event (manual or crash). This message will be received by the service operator and used to dispatch the appropriate emergency services.

Note the caller's phone number (i.e. number of the mobile unit) is available in the header and can be retrieved from there. This message is currently defined as 97 bytes of data. When the header is added, this rises to 145 bytes when using CRC-32 and FEC-15 encoding. There are some pieces of information more critical than others. Some of the less critical data can still be removed or the lower FEC-2 encoding can be used. It is currently believed that FEC-15 will only be required on BS26 to improve the reliability.

19.2.2 Message type

Message type is ECMsg. The ACP Message Type field is 0x06.

19.2.3 Message Structure and Content

Note the field definitions are defined as bits and begin on bit boundaries. For convenience some fields are padded with bit spares to force some fields to begin on byte boundaries. All bits within a byte are addressed from left to right as bit 0, 1, ..., 7. A field value of 0x80 in a byte refers to the left most bit in the byte and is bit 0. This is done in order to maintain consistency for bit fields that span multiple bytes.

The ECALL message is divided into 2 packets. The 2nd packet is only sent when the numbers of transport container numbers more than 4. When there are 4 or fewer containers specified, the SO receives an ECALL message in a single packet; the **Last Flag** bit will be set in the ACP header of the 1st packet and there will be no 2nd packet. When there are more than 4 containers specified, the SO receive an ECALL message with 2 packets; the **Last Flag** bit is set in the ACP header of the 2nd packet.

The **MID** field is used to define different formats for ECALL messages for different **TCU**s as well as version identification. The subfield Manufacturer ID consists of the leftmost 4 bits and the rightmost 4 bits concatenated into an 8-bit field that currently only values 0 and 1 are defined for MaufacturerID. This field along with the field **ACP Message Type** field provides for orderly growth of different message types and definitions for different product lines.

Table 1: Emergency Call Message (1st packet)

Field	Type	bits	Description
MID	uint8	2*8	byte 0: (bit 0..3): Manufacturer ID (part 1),

			byte 0: (bit 4..7) Major hardware release # byte1: (bit 0..3) Major software release # byte1: (bit 4..7) Manufacturer ID (part 2) (0x0 = Mercedes-Benz) (0x1 = Renault) (0x2 = Nokia)
CurrentRawPosition	uint8	20*8	byte 0-19 This is the raw current position from the GPS without any dead reckoning modifications. The form is compressed from the WGS 84 format sLLLLllllhhhh This is compressed from the form provided from the GPS device. bits 0-7 This is the time for this GPS measurement. It is the delta, in seconds, prior to the time this message was triggered. Thus it could be from 0 to 254 seconds prior to the timestamp in this message. If the time delta is 255 seconds or larger, then this field is set to 255. bits 8-37 LLLL: Latitude in msec bits 38-68 llll: Longitude in msec bits 69-77 reserved: set to 0 bits 78-109 hhhh bit 110-115 channel 1 satellite id bit 116-121 channel 2 satellite id bit 122-127 channel 3 satellite id bit 128-133 channel 4 satellite id bit 134-139 channel 5 satellite id bit 140-145 channel 6 satellite id bit 146-151 channel 7 satellite id bit 152-157 channel 8 satellite id If no GPS message data is available, this entire field shall be 0. However, the GPS time will be the GMT time of day. bit 158-159 time unit definition for bits 0-7 0: time is in seconds 1: reserved, not currently used

			2: reserved, not currently used 3: reserved, not currently used
PriorRawPosition	uint8	20*8	byte 0-19 This is the raw prior position from the GPS without any dead reckoning modifications. The form is compressed from the WGS 84 format bits 0-7 This is the time for this GPS measurement. It is the delta, in seconds, prior to the time in the field currentRawPosition. Thus it could be from 0 to 254 seconds prior to the timestamp in currentRawPosition. If the time delta is 255 seconds or larger, then this field is set to 255. bits 8-159 same definition as in currentRawPosition
CurrentPositionDR	uint8	8*8	byte 0-7 This form is extracted from the WGS84 reference ellipsoid. It is the currentPosition including the dead reckoning data. bits 0-30 LLLL: Latitude bits 31-61 llll: Longitude
PriorPositionDeltas	2 * uint8 array[18]	36*8	array of 18 entries; each 2-byte delta (array[x]) when applied to array[x-1] will provide the absolute location for array[x]. array[0] is delta applied to the CurrentPositionDR. bit 0-7 delta y north(+) / south (-) difference to last transmitted value in signed meters. max +/- 127 m bit 8-15 delta x east(+) / west(-) difference to last transmitted value in signed meters. max +/- 127 m
CurrentTime	uint8	4*8	The time this message was triggered. It is GMT time. It is defined as follows: bit 0-4: day of month, range (1..31) bit 5-8: month of year, range (1..12) bit 9-14: year (00 = 1990 01 = 1991 ranges up to 63 = 2053) bit 15-19: hours, range (0..23) bit 20-25: minutes, range (0..59)

			bit 26-31: seconds, range (0..59)
EmergencyCode	uint8	1*8	bit field defining the reason for this ECALL 0x01: Automatic trigger - MRS 0x02: Automatic trigger - roll 0x04: Manual trigger 0x08: Taxi Ecall Bit 6 & Bit 7 Number of Occupants 0 0 Minimum 1 person 0 1 Minimum 2 persons 1 0 Minimum 3 person 1 1 Minimum 4 person 0x40: Status of Speaker ON 0x80: Test message
CrashIntensity	uint8	1*8	0x80 Not Defined 0x40 Seat belt tightener triggered by rear crash 0x20 Front airbag triggering, second threshold 0x10 Triggering of a sidebag 0x08 Emergency call triggering by rear crash, second threshold 0x04 Crash triggered by roll sensor 0x03 Reserved, set to 0 0x02 Test message

Sensorcode	uint8	1*8	bit 0: fire bit 1: leakage bit 2-7: to be defined
TransportGoodNumberContainer1	uint8	36	bits 0-35: Form is nnnnqqqqu It is entered at the vehicle by the customer via a future MMI interface. bits 0-15: nnnn = the coded number that identifies the cargo the vehicle is carrying. In Europe this can be the Hommel transport good number which is a 4digit decimal number. bits 16-31 qqqq = the quantity of the items in nnnn above using units u. It is a 16 digit binary number with range (1..65535) bits 32-35 u = the units for the quantity qq persons (0x00) tons (0x01) liters (0x02) kilograms (0x03) cubic-meter (0x04) spares (0x05 - 0xF)
TransportGoodNumberContainer2	uint8	36	bits 0-35: Defined same as transportGoodNumberContainer1
TransportGoodNumberContainer3	uint8	36	bits 0-35: Defined same as transportGoodNumberContainer1
TransportGoodNumberContainer4	uint8	36	bits 0-35: Defined same as transportGoodNumberContainer1
Total message bytes		888 bits or	

Table 1: Emergency Call Message (2nd Packet)

Field	Type	bites	Description
TransportGoodNumber Container5	Uint8	36	bits 0-35: Defined same as transportGoodNumberContainer1
TransportGoodNumber Container6	Uint8	36	bits 0-35: Defined same as transportGoodNumberContainer1
TransportGoodNumber Container7	Uint8	36	bits 0-35: Defined same as transportGoodNumberContainer1
TransportGoodNumber Container8	Uint8	36	bits 0-35: Defined same as transportGoodNumberContainer1
TransportGoodNumber Container9	Uint8	36	bits 0-35: Defined same as transportGoodNumberContainer1
TransportGoodNumber Container10	Uint8	36	bits 0-35: Defined same as transportGoodNumberContainer1
Total message bytes		216 bits or	

19.2.4 Triggering Event

This message will be sent when the TCU has detected a crash or the manual push-button has been pressed.

In general if the botton on the TCU has been pressed; an ECall msg has been queued to be sent; but a crash indication arrives, the crash indication takes precedence. The ECall msg to be sent is the crash indication.

Also if the button on the TCU has been pressed; an ECall msg has been sent; then a crash indication arrives, the original ECall msg sequence continues to completion; then the Ecall msg for the crash indication is sent.

19.2.5 Responses

The service operator will respond to this message by sending an Emergency Call Confirmation Message to the TCU.

19.2.6 System Issues

SMS Transmission:

The SO will need to make sure that duplicate packets are handled. Since SMS is not a real-time mechanism, a response time-out can likely occur before the response packet can be sent. The SO shall acknowledge duplicate packets (will respond to them). The SO may discard the duplicate packet. The TCU shall recognize duplicate acknowledgments and discard the duplicates. If the TCU receives an acknowledgement for which a data packet no longer exists, the TCU shall discard the acknowledgement. The TCU expects an ACP acknowledgment as well as an application level ECCMsg for confirmation.

19.2.7 Emergency Call Confirmation Message (SO -> TCU)

19.2.8 General Description

The acknowledge mechanism of the ACP shall be used to return a EC confirmation response. Any necessary data can be piggybacked onto the acknowledge packet. The message type is 0x0E.

19.2.9 Message type

Message type is ECCMsg. ACPMessageType field is 0x0E.

19.2.10 Message Structure and Content (Data only)

The phone number of the operator is in the packet header.

Field	Type	bytes	Description
VoiceNumber	unit8	10	Up to 19 digits for the voice call to call the SO or PESC. (4 bit packed decimal ending with four 1’s)
ServiceCode	unit8	1	reserved, currently set to 0
TerminateECALL	uint8	1	0x00 = continue sending additional ECALL messages 0xFF = terminate sending ECALL messages The only time this field is 0xFF is when in taxi ECALL mode and the service operator wants the taxi to terminate sending ECALL messages at periodic intervals.
Confirmation	unit8	1	0x00 = denial 0x80 = acceptance --------------------------------------- bit 0x40 defines state of phone speaker for ecall 0x40 = on ecall switch speaker ON 0x00 = on ecall switch speaker OFF -- additional possible values 0xC0 = acceptance and switch speaker ON 0x80 = acceptance and switch speaker OFF
Total Message Bytes		13	

19.3 Triggering Event

When an Emergency Call Message has been received by the service operator this message will be sent to the TCU to confirm receipt.

19.4 Responses

There is no response for this message

19.5 System Issues

The SO should issue Emergency Call Confirmation Messages for duplicate emergency call messages received. Duplicate EC messages can be recognized by the same value in the MessageID field in the packet header. The TCU shall recognize the duplicate acknowledgment. The prior ECCMsg may not have been properly received, thus an ECCMsg is needed by the TCU. Duplicate ECCMsg must be handled by the TCU.

20 Appendix C: Correlated GPS Position Message Description

8-CHANNEL POSITION/STATUS/DATA OUTPUT MESSAGE

MOTOROLA BINARY FORMAT

This message is output at 1 Hz, once enabled. It provides the GPS-only solution for position as well as the integrated GPS/dead-reckoning solutions.

INPUT COMMAND DESCRIPTION:

² Set response message rate:

@@EoeC<CR><LF>

e - enable/disable output messages

0 - output response message
once (polled)

1..255 - response message
output at indicated rate
(continuous)
1 - once per second
2 - once every two seconds
255 - once every 255
seconds

C - checksum

Message Length: 8 bytes

RESPONSE MESSAGE:

² (1 Hz in response to enable command)

@@EomdyyhmsLLLLDDllllddhhhhsLLLLvvvvllllvvvvhhhheeeessssC<CR><LF>

Date	
m - month	1 .. 12
d - day	1 .. 31
yy - year	1980 .. 2079
Time	
h - hours	0 .. 23
m - minutes	0 .. 59
s - seconds	0 .. 60

GPS position solution and DOP

LLLL - -324,000,000

latitude in msec 324,000,000

(-90× to +90×)

DD -

NS DOP squared (0.01 resolution, 0-65535)

llll - -648,000,000

longitude in msec 648,000,000

(-180× to +180×)

dd -

EW DOP squared (0.01 resolution, 0-65535)

hhhh - height in cm -100,000

1,800,000

(-1000.00 to

+18,000.00 meters)

relative to WGS84

s - Solution type (0=no fix, 2=2d, 3=3D)

GPS/DR Integrated Solution

LLLLL latitude (milli-arcsecs - range +/- 3.240E+08)

vvvv NS pos. error variance (meters squared - 0.01 resolution)

llll longitude (milli-arcseconds range +/- 6.48E+08)

vvvv EW pos. error variance (meters squared - 0.01 resolution)
hhhh heading (radians - 0.0001 resolution)

eeee hdg error variance (radians squared, 0.01 resolution)

ssss speed (cm/sec - 1 cm/sec resolution)

C - checksum

Message Length: 59 bytes

21 Appendix D: Non-Correlated GPS Position Message Description

8-CHANNEL POSITION/STATUS/DATA OUTPUT MESSAGE

MOTOROLA BINARY FORMAT

This parameter determines the rate position/ status/data information (see Response Message) is output. The input mode parameter (m) instructs the GPS Receiver to either output this message one time (polled), or to output this message at the indicated update rate (continuously). Once the GPS Receiver is set to continuous output, the continuous message flow can be stopped by sending a one-time output request. The GPS Receiver will output the message one more time, then terminate any further message outputs. The state of the mode bit is stored in non-volatile memory. If the GPS Receiver was continuously outputting the position /status/data output when turned off, it will begin to output this message continuously (at the selected update rate) again when power is reapplied.

If the GPS Receiver has its Idle Mode option enabled, this parameter selection does not exist, and the GPS Receiver outputs the last known valid position/status/data message once when polled.

INPUT COMMAND DESCRIPTION:

² Set response message rate:

@@EamC<CR><LF>

m - mode 0 - output response message
once (polled)
1..255 - response message
output at indicated rate
(continuous)
1 - once per second
2 - once every two seconds
255 - once every 255
seconds

C - checksum

Message Length: 8 bytes

RESPONSE MESSAGE:

² (to command)

@@EamdyyhmsffffaaaaoooohhhhmmmmvvhhddtntimsdimsdimsdimsdimsdimsdimsdimsdsC<CR><LF>

Date

m - month	1 .. 12
d - day	1 .. 31

yy - year	1980 .. 2079	
Time		
h - hours	0 .. 23	
m - minutes	0 .. 59	
s - seconds	0 .. 60	
ffff - fractional sec	0 .. 999,999,999	
	(0.0 to 0.999999999)	
Position		
aaaa -	-324,000,000	
latitude in msec	324,000,000	
	(-90× to +90×)	
oooo -	-648,000,000	
longitude in msec	648,000,000	
	(-180× to +180×)	
hhhh - height in cm	-100,000	
	1,800,000	
	(-1000.00 to	
	+18,000.00 meters)	
mmmm - height in cm	-100,000	
	1,800,000	
	(-1000.00 to	
	+18,000.00 meters)	
Velocity		
vv - velocity in cm/sec	0 .. 51400	
	(0 to 514.00 m/sec)	
hh - heading	0 .. 3599	
	(true north res 0.1×)	(0.0 to 359.9 deg)

Note: 1 degree of position equals 3,600
sec. or 3,600,000 msec.

GPS Receiver reported height is limited to 18,000 meters, and velocity output is limited to 514.0 m/sec by United States export laws. If the GPS Receiver is used above these limits, the height and velocity outputs are clamped to the maximum values. In addition, the latitude and longitude data will be incorrect.

Geometry	
dd - current DOP	0 .. 999

	(0.0 to 99.9 DOP)
(0.1 res)	(0 - not computable),
	or position-hold, or
	position prop)
t - DOP type	0 - PDOP (in 3D)
	1 - HDOP (in 2D)

Satellite visibility and tracking status

n - num of visible sats	0 .. 12
t - num of satellites tracked	0 .. 8

For each of eight receiver channels

i - sat ID	0 .. 37
m - channel tracking mode	0 .. 8

0 - Code Search

1 - Code Acquire

2 - AGC Set

3 - Freq Acquire

4 - Bit Sync Detect

5 - Message Sync Detect

6 - Satellite Time Avail

7 - Ephemeris Acquire

8 - Avail for Position

s - Signal Strength 0 .. 255

(number proportional to SNR)

d - Channel Status Flag

Each bit represents one of the following:

(msb) Bit 7: Using for Position Fix

Bit 6: Satellite Momentum Alert Flag

Bit 5: Satellite Anti-Spoof Flag Set

Bit 4: Satellite Reported Unhealthy

Bit 3: Satellite Reported Inaccurate

(> 16 meters)

Bit 2: Spare

Bit 1: Spare

(lsb) Bit 0: Parity Error

(End of Channel Dependent Data)

s - Receiver Status Message

(msb) Bit 7: Position Propagate mode

Bit 6: Poor Geometry (DOP > 20)

Bit 5: 3D fix

Bit 4: Altitude Hold (2D fix)

Bit 3: Acquiring Satellites/Position hold

Bit 2: Differential

Bit 1: Insufficient visible satellites (<3)

(lsb) Bit 0: Bad Almanac

C - checksum

Message Length: 76 bytes