

BS ISO 13185-2:2015



BSI Standards Publication

Intelligent transport systems — Vehicle interface for provisioning and support of ITS services

Part 2: Unified gateway protocol (UGP)
requirements and specification for vehicle
ITS station gateway (V-ITS-SG) interface

bsi.

...making excellence a habit.
AVA Documents™
Document Sharing Hub

National foreword

This British Standard is the UK implementation of ISO 13185-2:2015.

The UK participation in its preparation was entrusted to Technical Committee EPL/278, Intelligent transport systems.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2015.
Published by BSI Standards Limited 2015

ISBN 978 0 580 81820 2
ICS 03.220.01; 35.240.60

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 April 2015.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

**Intelligent transport systems —
Vehicle interface for provisioning and
support of ITS services —**

**Part 2:
Unified gateway protocol (UGP)
requirements and specification for
vehicle ITS station gateway (V-ITS-
SG) interface**

*Systèmes intelligents de transport — Interface véhicule pour la
fourniture et le support de services ITS —*

*Partie 2: Exigences de protocole et spécification pour l'interface
passerelle de la station ITS du véhicule*





COPYRIGHT PROTECTED DOCUMENT

© ISO 2015

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword	vi
Introduction	vii
1 Scope	1
2 Normative references	1
3 Terms, definitions, symbols, and abbreviated terms	1
3.1 Terms and definitions	1
3.2 Abbreviated terms	2
4 Conventions	3
5 Document overview	4
6 UGP application layer services	6
6.1 General	6
6.2 Service primitives	6
6.3 Application layer service primitives — Confirmed service	7
6.4 Format description of service primitives	8
6.4.1 General definition	8
6.4.2 Service request and service indication primitives	8
6.4.3 Service response and service confirm primitives	9
6.4.4 Service request-confirm and service response-confirm primitives	10
6.5 UGP service call with global reply message handling	10
6.6 UGP service call with scheduled-based reply message handling	11
6.7 UGP service data unit specification	12
6.7.1 Mandatory parameters	12
7 UGP application layer protocol in ASN.1 document interface	13
7.1 General definition	13
7.2 Common message data and structure	13
7.3 ASN.1 model	14
8 Unified gateway protocol (UGP) clusters	15
8.1 Overview	15
8.2 UGP service clusters and associated services	15
9 UGP service cluster 1 — Global services	16
9.1 GlobalPositiveReply	16
9.2 GlobalNegativeReply	16
9.3 StopService	17
9.3.1 Service description	17
9.3.2 Message 'StopServiceCall'	17
9.3.3 Positive reply	17
9.3.4 Error handling	17
9.3.5 Example	17
9.4 Reset	18
9.4.1 Service description	18
9.4.2 Message 'ResetCall'	18
9.4.3 Positive reply	18
9.4.4 Error handling	18
9.4.5 Example	18
10 UGP service cluster 2 — Authentication	19
10.1 Authentication	19
10.1.1 Service description	19
10.1.2 Message 'AuthenticationCall'	19
10.1.3 Message 'AuthenticationReply'	19
10.1.4 Error handling	20

10.1.5	Example	20
11	UGP service cluster 3 — Supported data	21
11.1	GetSupportedData	21
11.1.1	Service description	21
11.1.2	Message ‘GetSupportedDataCall’	21
11.1.3	Message ‘GetSupportedDataReply’	22
11.1.4	Error handling	23
11.1.5	Example without ECU data	23
11.1.6	Example with ECU data	24
11.1.7	ECU filtered example	25
11.1.8	Negative access type and data parameter property filtered example	26
12	UGP service cluster 4 — Data parameter access	26
12.1	GetValue	26
12.1.1	Service description	26
12.1.2	Message ‘GetValueCall’	26
12.1.3	Message ‘GetValueReply’	27
12.1.4	Error handling	28
12.1.5	Single get data parameter list example	28
12.1.6	Asynchronous conditioned example	29
12.1.7	Monitor example	30
12.1.8	Structure example ‘vehicle info’	31
12.2	SetValue	32
12.2.1	Service description	32
12.2.2	Message ‘SetValueCall’	32
12.2.3	Positive reply	33
12.2.4	Error handling	33
12.2.5	Example	33
12.3	ControlValue	34
12.3.1	Service description	34
12.3.2	Message ‘ControlValueCall’	34
12.3.3	Message ‘ControlValueReply’	34
12.3.4	Error handling	35
12.3.5	Example 1	35
12.3.6	Example 2	38
13	UGP service cluster 5 — Diagnostic trouble code information access	39
13.1	GetDtcInfo	39
13.1.1	Service description	39
13.1.2	Message ‘GetDtcInfoCall’	39
13.1.3	Message ‘GetDtcInfoReply’	40
13.1.4	Error handling	41
13.1.5	Examples	41
13.2	ClearDtcInfo	42
13.2.1	Service description	42
13.2.2	Message ‘ClearDtcInfoCall’	42
13.2.3	Positive reply	42
13.2.4	Error handling	42
13.2.5	Example	42
14	UGP service cluster 6 — In-vehicle network access	43
14.1	EnablePassThru	43
14.1.1	Service description	43
14.1.2	Message ‘EnablePassThruCall’	43
14.1.3	Positive reply	44
14.1.4	Error handling	44
14.1.5	Example	44
15	UGP service cluster 7 — Maintenance	45
15.1	ListFile	45

15.1.1	Service description	45
15.1.2	Message 'ListFileCall'	45
15.1.3	Message 'ListFileReply'	46
15.1.4	Error handling	46
15.1.5	Example	46
15.2	ManageFile	47
15.2.1	Service description	47
15.2.2	Message 'ManageFileCall'	47
15.2.3	Message 'ManageFileReply'	48
15.2.4	Error handling	48
15.2.5	Example download	48
15.2.6	Example upload	49
15.2.7	Example delete	50
Annex A (normative) Vehicle Interface Data Format definition (VIDF)		51
Annex B (normative) Unified gateway protocol ASN.1 definition		70
Bibliography		78

Foreword

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

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

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

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

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: [Foreword - Supplementary information](#)

The committee responsible for this document is ISO/TC 204, *Intelligent transport systems*.

ISO 13185 consists of the following parts, under the general title *Intelligent transport systems (ITS) — Vehicle interface for provisioning and support of ITS services*:

- *Part 1: General information and use case definition*
- *Part 2: Unified gateway protocol (UGP) requirements and specification for vehicle ITS station gateway (V-ITS-SG) interface*

The following parts are under preparation:

- *Part 3: Unified gateway protocol (UGP) server and client API specification*
- *Part 4: Unified gateway protocol (UGP) conformance test specification*

Introduction

This part of ISO 13185 has been established to define the requirements of a common software interface to a vehicle gateway to easily exchange vehicle information data amongst nomadic and/or mobile device, vehicle gateway, and the vehicle's Electronic Control Units (ECUs).

Applications supporting service provision use via nomadic and mobile devices need vehicle information data through an in-vehicle interface access method, as well as the harmonization of existing standards to support a single vehicle data access solution.

This document defines an ASN.1-based protocol between the nomadic and/or mobile device (ND) and the UGP Server (implemented in the V-ITS-SG) in the vehicle.

To achieve this, it is based on the Open Systems Interconnection (OSI) Basic Reference Model specified in ISO/IEC 7498-1 and ISO/IEC 10731, which structures communication systems into seven layers.

This part of ISO 13185 can be used by vehicle manufacturers for future vehicle design to support the design of vehicle gateways to interface with NDs.

The ND applications need vehicle information data through an in-vehicle interface access method (V-ITS-S with V-ITS-SG).

This part of ISO 13185 supports ITS applications which are based on ND in vehicles to operate on a common software interface to a V-ITS-SG to easily exchange vehicle information data among ND, vehicle V-ITS-SG, and ECUs.

The protocol implementation in the vehicle gateway features the following:

- the deny of access to the vehicle gateway data by unauthorized on-board and off-board test equipment;
- the deny of access to parts of the vehicle gateway data by unauthorized on-board and off-board test equipment (privacy);
- the identification of the vehicle gateway and the vehicle it is installed in;
- the list of in-vehicle connected ECUs to the vehicle gateway and their data parameters;
- methods to configure the access to vehicle data.

Intelligent transport systems — Vehicle interface for provisioning and support of ITS services —

Part 2:

Unified gateway protocol (UGP) requirements and specification for vehicle ITS station gateway (V-ITS-SG) interface

1 Scope

This part of ISO 13185 specifies the requirements of an ASN.1-based protocol between a vehicle-ITS-Station Gateway (V-ITS-SG) and a nomadic and/or mobile device (ND) to easily exchange vehicle information data.

The ASN.1-based protocol has been specified to support a wired or wireless connection between the ND and V-ITS-SG.

The Unified Gateway Protocol (UGP) is used between the V-ITS-SG and the ND. UGP supports several features in order to provide

- authorization (data privacy),
- secured access,
- V-ITS-SG and in-vehicle ECUs identification,
- real-time vehicle data parameters with identifier and type information in ASN.1 format, and
- enhanced vehicle data parameters with identifier and type information in ASN.1 format.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 13185-1, *Intelligent transport systems — Vehicle interface for provisioning and support of ITS services — Part 1: General information and use case definition*

ISO 14229-2, *Road vehicles — Unified diagnostic services (UDS) — Part 2: Session layer services*

3 Terms, definitions, symbols, and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 13185-1 and the following apply.

3.1.1 authentication

cryptographic service that provides assurance that the sender of a communication is who they claim to be

Note 1 to entry: Authentication might involve confirming the identity of a person or software program, tracing the origins of an artefact, ensuring that a product is what its packaging and labelling claims to be.

3.1.2 authorization

right granted by an authority

3.1.3 application layer protocol data unit A_PDU

group of information added or removed by a layer of the OSI model

Note 1 to entry: Each layer in the model uses the PDU to communicate and exchange information, which can only be read by the peer layer on the receiving device and is then handed over to the next upper layer after stripping.

Note 2 to entry: The A_PDU (Application layer Protocol Data Unit) is directly constructed from the A_SDU (Application layer Service Data Unit) and the layer specific control information A_PCI (Application layer Protocol Control Information).

3.1.4 application layer service data unit A_SDU

unit of data that has been passed down from an OSI layer to a lower layer and that has not yet been encapsulated into a protocol data unit (PDU) by the lower layer

Note 1 to entry: It is a set of data that is sent by a user of the services of a given layer and is transmitted semantically unchanged to a peer service user. A_SDU is the SDU in the OSI layer 7 (application layer).

3.1.5 privacy

choice made by the vehicle owner to grant information access (refer to directive 2002/58/EC dated 12 July 2002).

3.1.6 UGP Client

client implementing the UGP services

Note 1 to entry: For example, ND.

3.1.7 UGP Server

server implementing the UGP services

Note 1 to entry: For example, V-ITS-SG.

3.1.8 unified gateway protocol UGP

application layer protocol to enable a *UGP Client* ([3.1.6](#)) to access data from the *UGP Server* ([3.1.7](#))

3.2 Abbreviated terms

API	application programming interface
ASN.1	abstract syntax notation one
A_PDU	application layer protocol data unit

API	application programming interface
A_SDU	application layer service data unit
BT	bluetooth
C	conditional
C-ITS-S	central-intelligent transport system-station
CRC	cyclic redundancy check
Cvt	convention (M, O, C)
DTC	diagnostic trouble code
ECU	electronic control unit
ITS	intelligent transport systems
HCI	host controller interface
L2CAP	logical link control and adaptation protocol
LC	baseband link controller
LMP	link manager protocol
M	mandatory
ND	nomadic device
O	optional
OBEX	object exchange
PER	packed encoding rules
P-ITS-S	personal- intelligent transport system-station
RFCOMM	radio frequency communication (serial emulation API)
R-ITS-S	roadside- intelligent transport system-station
SDP	service discovery protocol
UDS	unified diagnostic services
UGP	unified gateway protocol
U-PER	unaligned packed encoding rules
VIN	vehicle identification number
V-ITS-SG	vehicle-intelligent transport system-station gateway

4 Conventions

This part of ISO 13185 is based on the conventions discussed in the OSI Service Conventions (ISO/IEC 10731:1994) as they apply for communication services.

These conventions specify the interactions between the service user and the service provider. Information is passed between the service user and the service provider by service primitives, which can convey parameters.

The distinction between service and protocol is summarized in [Figure 1](#).

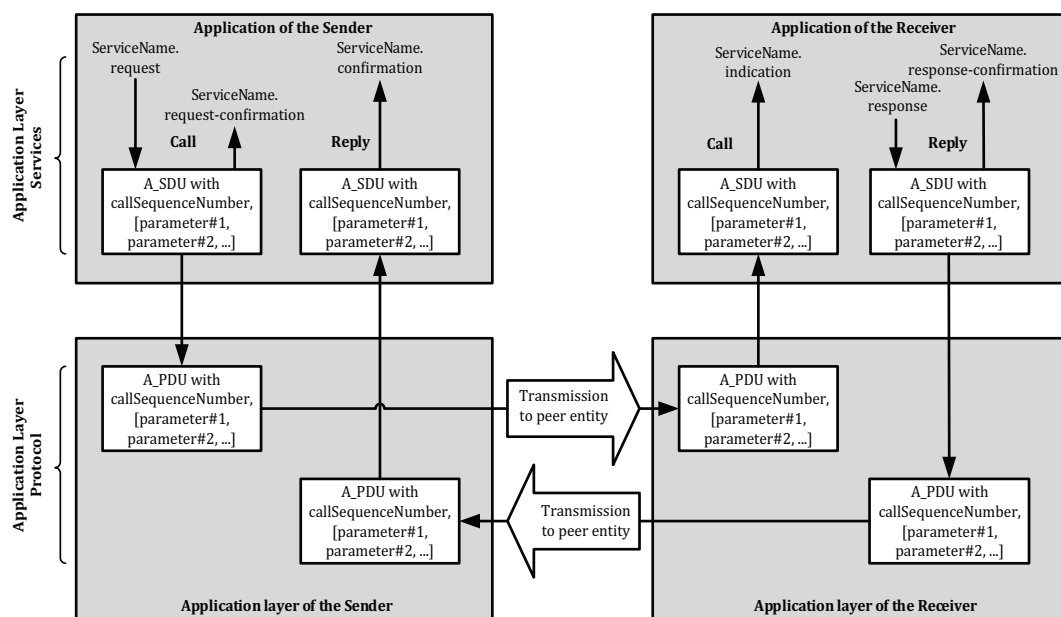


Figure 1 — Services and the protocol

This part of ISO 13185 defines services using the six service primitives: request, req_confirm, indication, response, rsp_confirm, and confirmation.

The request and indication service primitives always have the same format and parameters. Consequently for all services, the response and confirmation service primitives (except req_confirm and rsp_confirm) always have the same format and parameters. When the service primitives are defined in this part of ISO 13185, only the request and response service primitives are listed.

5 Document overview

[Figure 2](#) shows the UGP document structure and automotive protocols and interfaces.

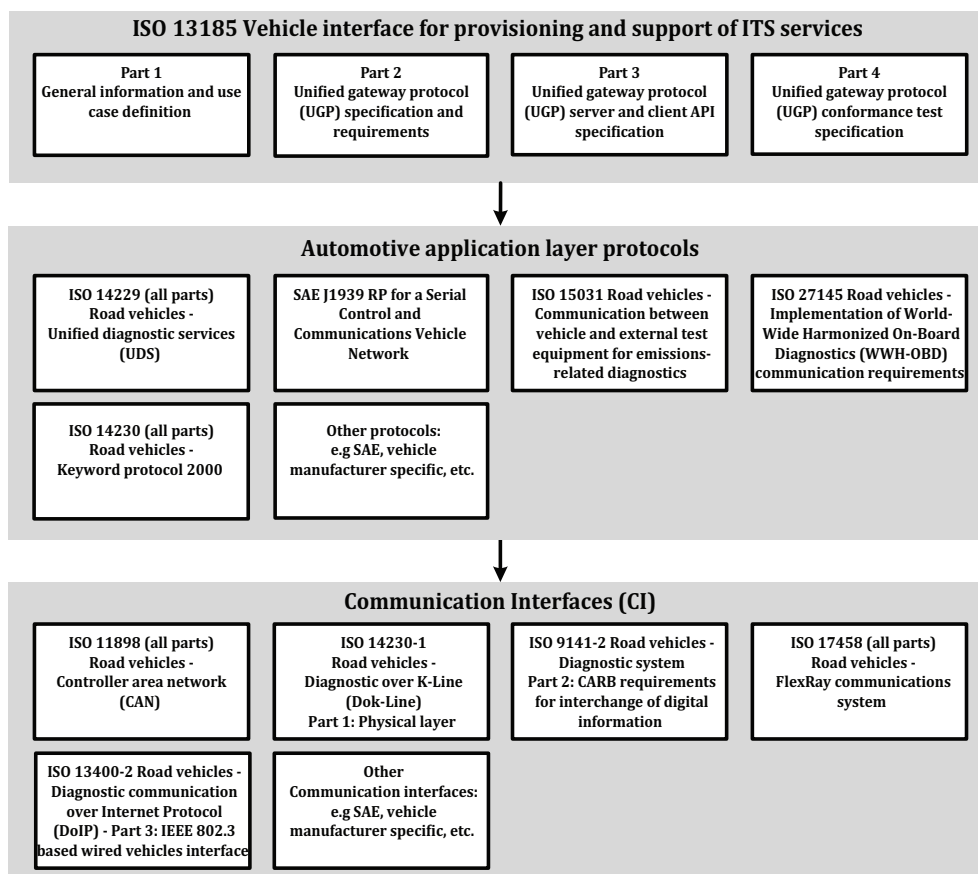


Figure 2 — UGP document structure and automotive protocols and interfaces

[Figure 3](#) illustrates the implementation of the UGP and UDS document reference according to OSI model into a V-ITS-SG. Both UGP and UDS use the same session layer services as defined in ISO 14229-2. Non-UDS based protocols use their own session, transport, and network layer services.

Communication services as they apply to the OSI Service Conventions:

- **Application layer:**
This part of ISO 13185 specifies the requirements of an ASN.1 based UGP independent of underlying data link.
- **Session layer:**
ISO 14229-2 specifies the session layer services. This part of ISO 13185 uses the same session layer services as ISO 14229-1.
- **Transport and Network layer:**
Each communication link provides its own set of transport protocols and network layer services.
- **Data link and Physical layer:**
Each communication link provides its own set of data link and physical layer services.

The OSI layered concept implemented for the ISO 14229 series is the same as the one used for the ISO 13185 series.

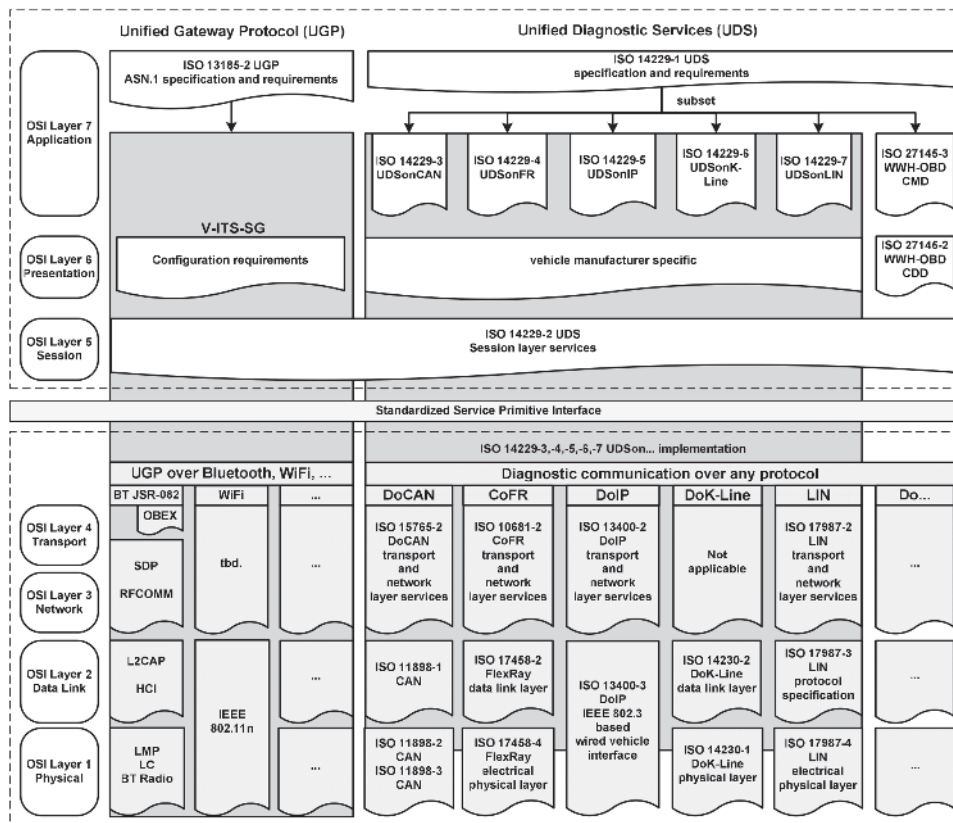


Figure 3 — Implementation of UGP document reference according to OSI model

6 UGP application layer services

6.1 General

Figure 4 illustrates the connection between the ND and V-ITS-SG. UGP provides application layer services to exchange the data between the Nomadic Device and the V-ITS-SG.

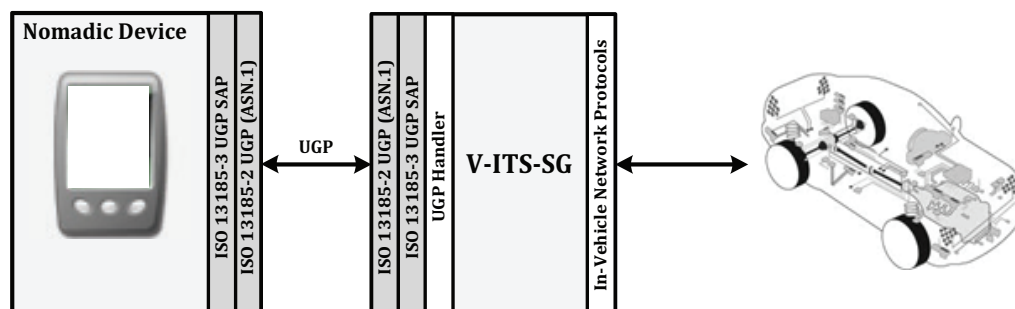


Figure 4 — Connection between ND and V-ITS-SG

6.2 Service primitives

Application layer services are referred to as Unified Gateway Protocol (UGP) services. The application layer services are used in client-server based systems to perform functions such as test, inspection, monitoring or diagnosis of on-board vehicle servers. The UGP Client, referred to as the P-ITS-S, R-ITS-S, or C-ITS-S, uses the application layer UGP services to request data from the UGP Server. The UGP Server uses the application layer services to send response data, provided by the requested UGP service, back

to the UGP Client. The usage of application layer UGP services is independent from the UGP Client being an off-board or on-board tester. It is possible to have more than one UGP Client connected.

The UGP server and client API provides a number of services that all have the same general structure. For each service, six service primitives are specified:

- a **service request primitive**, used by the client function in the UGP application, to pass data about a requested UGP service to the diagnostics application layer;
- a **service request-confirmation primitive**, used by the client function in the UGP application, to indicate that the data passed in the service request primitive is successfully sent on the V-ITS-SG the nomadic device is connected to;
- a **service indication primitive**, used by the UGP application layer, to pass data to the server function of the server UGP application;
- a **service response primitive**, used by the server function in the V-ITS-SG UGP application, to pass response data provided by the requested UGP service to the UGP application layer;
- a **service response-confirmation primitive**, used by the server function in the V-ITS-SG UGP application, to indicate that the data passed in the service response primitive is successfully sent on the V-ITS-SG the nomadic device received the UGP request on;
- a **service confirmation primitive** used by the UGP application layer to pass data to the client function in the UGP application.

6.3 Application layer service primitives — Confirmed service

For a given service, the request-confirmation primitive and the response-confirmation primitive always have the same service data unit. The purpose of these service primitives is to indicate the completion of an earlier request or response service primitive invocation.

The response-confirm primitive is used by the application layer to indicate an internal event, which is significant to the server application, and pass communication results of an associated previous service response to the server function in the V-ITS-SG application.

Figure 5 depicts the application layer service primitives – Confirmed service.

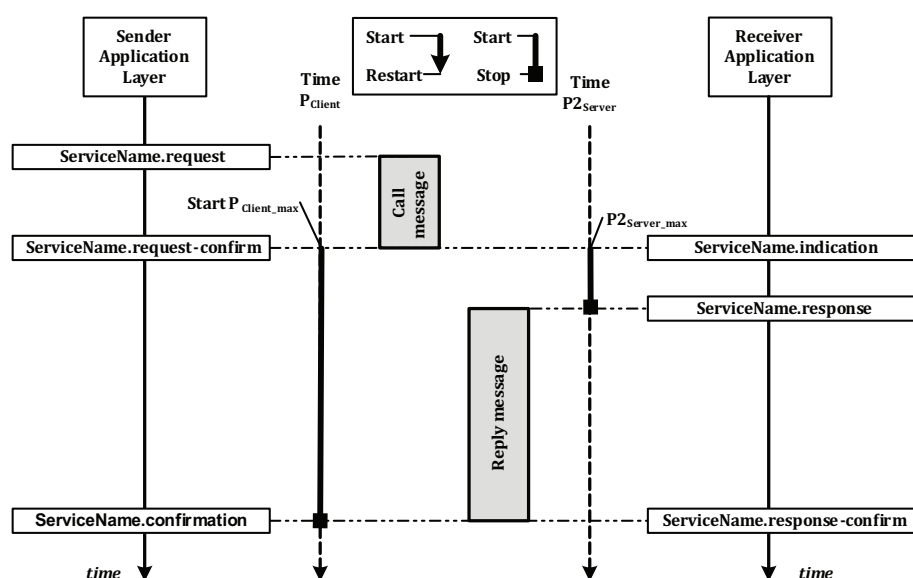


Figure 5 — Application layer service primitives – Confirmed service

6.4 Format description of service primitives

6.4.1 General definition

All application layer services have the same general format. Service primitives are written in the form:

```
service_name.type  
    (  
        parameter A, parameter B, parameter C  
        [,parameter 1, ...]  
    )
```

where

“service_name”	is the name of the UGP service (e.g. GetSupported-Data),
“type”	indicates the type of the service primitive (e.g. request),
“parameter A, ...”	is the A_SDU (Application layer Service Data Unit) as a list of values passed by the service primitive (addressing information),
“parameter A, parameter B, parameter C”	are mandatory parameters that shall be included in all service calls,
“[,parameter 1, ...]”	are parameters that depend on the specific service (e.g. parameter 1 can be the GetSupportedData service). The brackets indicate that this part of the parameter list can be empty.

6.4.2 Service request and service indication primitives

For each application layer service, service request and service indication primitives are specified according to the following general format:

```
service_name.request    (  
    callSequenceNumber,  
    timeInMillis,  
    length, data[, parameter 1,  
    ...],)
```

The request primitive is used by the client function in the application to initiate the service and pass data about the requested UGP service to the application layer.

```
service_name.indication (  
    callSequenceNumber,  
    timeInMillis, length,  
    data[, parameter 1, ...],  
)
```

The indication primitive is used by the application layer to indicate an internal event which is significant to the UGP Server application and pass data about the requested UGP service to the UGP Server function of the V-ITS-SG application.

The request and indication primitive of a specific application layer service always have the same parameters and parameter values. This means, that the values of individual parameters shall not be changed by the communicating peer protocol entities of the application layer when the data are transmitted from the UGP Client to the UGP Server. The same values that are passed by the client function in the UGP Client application to the application layer in the service request call shall be received by the UGP Server function of the application from the service indication of the peer application layer.

6.4.3 Service response and service confirm primitives

For each application layer service, service response and service confirm primitives are specified according to the following general format:

```
service_name.response    (
    callSequenceNumber,
    timeInMillis, length,
    data[, parameter 1, ...],
)
```

The response primitive is used by the server function in the V-ITS-SG application to initiate the service and pass response data provided by the requested UGP service to the application layer.

```
service_name.confirm    (
    callSequenceNumber,
    timeInMillis,
    length,data[, parameter 1,
    ...],)
```

The confirm primitive is used by the application layer to indicate an internal event which is significant to the UGP Client application and pass results of an associated previous service request to the client function in the e.g. P-ITS-S application. It does not necessarily indicate any activity at the remote peer interface, e.g. if the requested service is not supported by the UGP Server or if the communication is broken.

The response and confirm primitive of a specific application layer service always have the same parameters and parameter values. This means that the values of individual parameters shall not be changed by the communicating peer protocol entities of the application layer when the data are transmitted from the UGP Server to the UGP Client. The same values that are passed by the server function of the V-ITS-SG application to the application layer in the service response call shall be received by the client function in the e.g. P-ITS-S application from the service confirmation of the peer application layer.

For each response and confirm primitive two different service data units (two sets of parameters) will be specified as the following:

- A positive response and positive confirm primitive shall be used with the first service data unit if the requested UGP service could be successfully performed by the server function in the V-ITS-SG;
- A negative response and confirm primitive shall be used with the second service data unit if the requested UGP service failed or could not be completed in time by the server function in the V-ITS-SG.

6.4.4 Service request-confirm and service response-confirm primitives

For each application layer service, service request-confirm and service response-confirm primitives are specified according to the following general format:

```
service_name.req_confirm      (  
                                callSequenceNumber,  
                                timeInMillis,  
                                result  
                                )
```

The request-confirm primitive is used by the application layer to indicate an internal event, which is significant to the UGP Client application, and pass communication results of an associated previous service request to the client function in the e.g. P-ITS-S application.

```
service_name.rsp_confirm      (  
                                callSequenceNumber,  
                                timeInMillis,  
                                result  
                                )
```

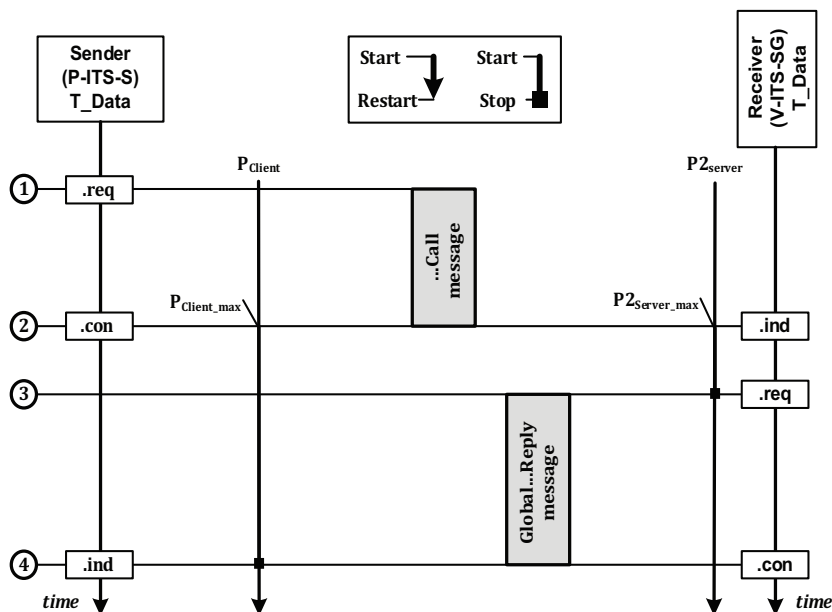
6.5 UGP service call with global reply message handling

The global reply message handling consists of the GlobalPositiveReply and GlobalNegativeReply messages. The UGP Server uses a GlobalNegativeReply message to inform a UGP client that the requested information is not accepted or supported. The GlobalPositiveReply is used by the UGP Server to positively respond to all UGP client calls without a call-specific reply message. These UGP services do not submit any data in the positive reply message.

[Figure 6](#) shows the UGP service call with global reply message handling.

The following assumptions apply:

- The UGP Client is i.e. a P-ITS-S and the UGP Server is the V-ITS-SG.
- The T_Data interface for the UGP Client (P-ITS-S) and the UGP Server (V-ITS-SG) is specified in ISO 14229-2. It provides a data link independent interface to the session and application layer services.
- The `callMode` parameter is set to `normal`.
- The `refreshInterval` is set to the default value 0.



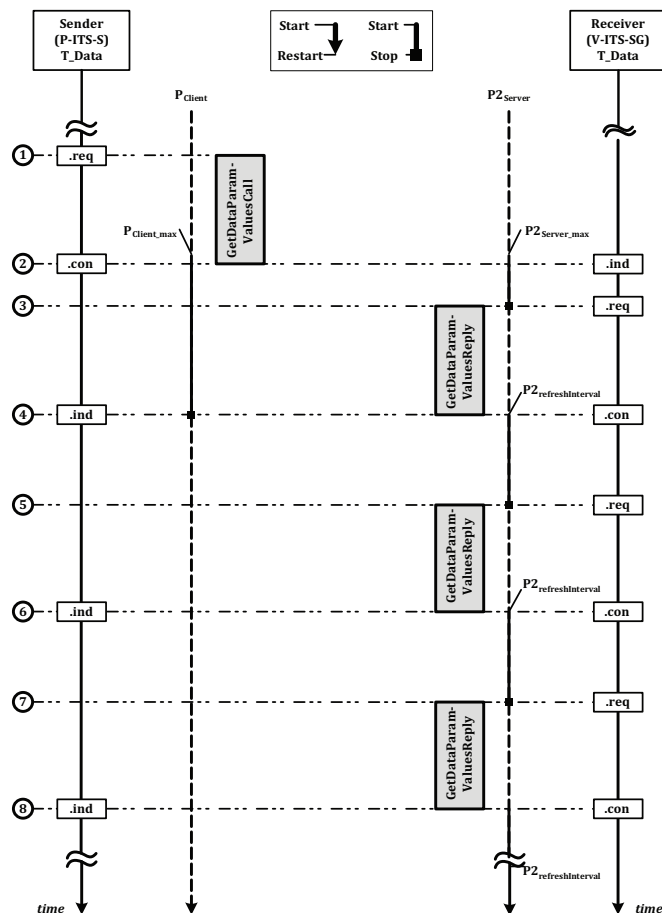
Key

- 1 **UGP Client (P-ITS-S) T_Data.req:** The i.e. P-ITS-S application of the UGP Client starts the transmission of a ...
Call message by issuing a T_Data.req to its communication layer. The communication layer transmits the ...Call
message to the UGP Server (V-ITS-SG).
- 2 **UGP Client (P-ITS-S) T_Data.con:** The completion of the ...Call message is indicated in the UGP Client
(P-ITS-S) via T_Data.con. Now the response timing as described in ISO 14229-2 applies.
UGP Server (V-ITS-SG) T_Data.ind: The completion of the UGP Client (P-ITS-S) ...Call message is indicated in
the UGP Server (V-ITS-SG) via the T_Data.ind. Now the response timing as described in ISO 14229-2 applies.
- 3 **UGP Server (V-ITS-SG) T_Data.req:** It is assumed that the UGP Client (P-ITS-S) requires a reply from the UGP
Server (V-ITS-SG). The UGP Server (V-ITS-SG) shall transmit the ...Reply message to indicate that the ...Call has
been processed and that the transmission of the reply has started.
- 4 **UGP Server (V-ITS-SG) T_Data.con:** The completion of the transmission of the ...Reply message is indicated
in the UGP Server (V-ITS-SG) via T_Data.con.

Figure 6 — UGP service call with global reply message handling

6.6 UGP service call with scheduled-based reply message handling

Figure 7 shows the UGP service call with scheduled-based reply message handling.



Key

- 1 **UGP Client (P-ITS-S) T_Data.req:** application issues a GetValueCall message to the transport/network layer.
- 2 **UGP Server (V-ITS-SG) T_Data.ind:** transport/network layer issues to application the indication of the reception of a GetValueCall message.
UGP Client (P-ITS-S) T_Data.con: transport/network layer issues to application the completion of the GetValueCall message. UGP Client starts its P_{Client} timer using the default reload value $P_{Client} = P_{Client_max}$. The value of the P_{Client} timer shall consider any latency that is involved based on the communication media between UGP Server and UGP Client.
- 3/5/7 **UGP Server (V-ITS-SG) T_Data.req:** application has prepared the GetValueReply message and issues a T_Data.req to transport/network layer within $P2_{Server_max}$. UGP Server stops the $P2_{Server}$ timer.
- 4/6/8 **UGP Server (V-ITS-SG) T_Data.con:** transport/network layer issues to application the completion of the GetValueReply message. UGP Server starts the $P2_{Server}$ timer using the value of $P2_{Server} = P2_{refreshInterval}$.
UGP Client (P-ITS-S) T_Data.ind: transport/network layer issues to application the reception of a GetValueReply message. UGP Client stops the P_{Client} timer.

Figure 7 — UGP service call with event- and scheduled-based reply message handling

6.7 UGP service data unit specification

6.7.1 Mandatory parameters

6.7.1.1 General definition

The application layer services contain three mandatory parameters. The following parameter definitions are applicable to all application layer services specified in this part of ISO 13185.

6.7.1.2 result

Type: enumeration

Range: ok, error

Description:

The parameter 'result' is used by the req_confirm and rsp_confirm primitives to indicate if a message has been transmitted correctly (ok) or whether the message transmission was not successful (error).

6.7.1.3 length

Type: 4 byte unsigned integer value

Range: $0_d - (2^{32}-1)_d$

Description:

This parameter includes the length of data to be transmitted/received.

6.7.1.4 data

This parameter includes all data to be exchanged by the higher layer entities.

6.7.1.5 callSequenceNumber

This parameter includes the sequence number of the call. The receiver shall use the same number for the reply.

6.7.1.6 timeInMillis

This parameter includes the time stamp of the reply data in milliseconds since 1970.

7 UGP application layer protocol in ASN.1 document interface

7.1 General definition

The communication between the UGP Client e.g. ND and the UGP Server (V-ITS-SG) is a unified gateway protocol with a high level of data abstraction. No knowledge about the vehicle protocols shall be required by the connected UGP Client.

The UGP Client application is pre-loaded with a vehicle-specific configuration in order to authenticate and interpret the data retrieved via the V-ITS-SG. The UGP Client displays or processes the requested information or relays the information to any kind of back office infra-structure, i.e. C-ITS-S. All information shall be composed of the V-ITS-SG and delivered to the UGP Client upon request if authentication and authorization criteria have been met by the UGP Client.

7.2 Common message data and structure

A UGP service consists of a request and response message. The data transmitted in the request and response message are structured in ASN.1. The data are encoded in Unaligned Packed Encoding Rules (U-PER). To distinguish the messages, all request messages end with 'Call' and all response messages end with 'Reply'.

Table 1 — Definition of UGPMessage

Msg	UGPMessage		
Attributes	Name	Description	Cvt
	callSequenceNumber	This parameter includes the sequence number of the call. The receiver shall use the same number for the reply.	M
	timeInMillis	This parameter includes the time stamp of the reply data in milliseconds since 1970.	0
ASN.1	<pre> UGPMessage ::= SEQUENCE { callSequenceNumber UNUM16, timeInMillis UNUM64 OPTIONAL, choiceUGP CHOICE { authenticationCall AuthenticationCall, authenticationReply AuthenticationReply, getSupportedDataCall GetSupportedDataCall, ... globalPositiveReply GlobalPositiveReply, globalNegativeReply GlobalNegativeReply, ... }, ... } ugpVersion Version ::= 1 </pre>		

All ASN.1 messages have a common basic structure. The root element is called UGPMessage. The message and its attributes and the ASN.1 representations are defined in [Table 1](#).

All services that have no positive response attributes use the global reply message GlobalPositiveReply (see [9.1](#)) instead of an own reply message.

All services that cannot support, interpret, etc. the attributes included in the message call (request) shall send the negative response message GlobalNegativeReply (see [9.2](#)).

7.3 ASN.1 model

The ASN.1 model consists of two modules:

- VehicleInterfaceDataFormat.asn: defines simple types like Boolean, String, UNUM16 and SNUM32. Additionally SEQUENCES for the data parameter definition are introduced: DisplayName, DataType, Numeric, EnumStringItem, DataParam, DataParamValue, etc. (see [B.1](#));
- UnifiedGatewayProtocol.asn: contains the UGPMessage with all of its choiceUGP like authenticationCall, getSupportedDataCall or getDtcInfoReply and the VIDFConfig (see [B.2](#)).

The ASN.1 model shall be used for validating the exchanged documents for a better understanding of the allowed structure.

8 Unified gateway protocol (UGP) clusters

8.1 Overview

[Table 2](#) provides an overview of the UGP clusters. A UGP cluster can have one or more UGP services.

Table 2 — UGP clusters and associated services

# — Main title of cluster	Brief description	Related services	Cvt
1 — Global services	This cluster handles the positive response for all UGP services without response parameters and all negative responses. Asynchronous services can be stopped and the key can be reset.	9.1 GlobalPositiveReply	—
		9.2 GlobalNegativeReply	—
		9.3 StopService	M
		9.4 Reset	M
2 — Authentication	The UGP services belonging to this cluster initiate the communication to the V-ITS-SG while checking the authentication and authorization.	10.1 Authentication	O ^a
3 — Supported data	The single UGP service belonging to this cluster requests the supported data parameters. This UGP service should be called once after authentication.	11.1 GetSupportedData	M ^b
4 — Data parameter access	The UGP services belonging to this cluster get, set and control the data parameter values in the V-ITS-SG (UGP Server). Getting and controlling can be scheduled in real time intervals by the V-ITS-SG.	12.1 GetValue	M
		12.2 SetValue	C ^c
		12.3 ControlValue	C ^c
5 — Diagnostic trouble code information access	The UGP services belonging to this cluster handle the reading and clearing of the diagnostic trouble codes and associated information.	13.1 GetDtcInfo	M
		13.2 ClearDtcInfo	M
6 — In-vehicle network access	The UGP services belonging to this cluster enable a switch to another protocol in the V-ITS-SG (UGP Server), which the vehicle manufacturer might choose to perform non-public configurations e.g. required during the assembly of the vehicle.	14.1 EnablePassThru	C ^c
7 — Maintenance	The UGP services belonging to this cluster maintain the V-ITS-SG (server) core software and its configuration with services to list, download, upload and delete. In addition, log files and snapshots can be listed and downloaded from the V-ITS-SG.	15.1 ListFile	C ^c
		15.2 ManageFile	C ^c
^a The Authentication service is optional. If the service is not executed, the default access rights are active (see 10.1).			
^b The GetSupportedData service shall only report the default data parameters which are required by local legislation and appropriate for the vehicle type and model, if the Authentication service has not been executed.			
^c This service is conditional and depends on the user access rights of the AuthenticationReply. If the Authentication service has not been executed, a GlobalNegativeReply is returned from the V-ITS-SG.			

8.2 UGP service clusters and associated services

[Figure 8](#) shows the UGP service clusters and associated services. The services shown in cluster “authentication and encryption” and “identification and supported data” shall be executed in the sequence as shown prior to any services of the other clusters.

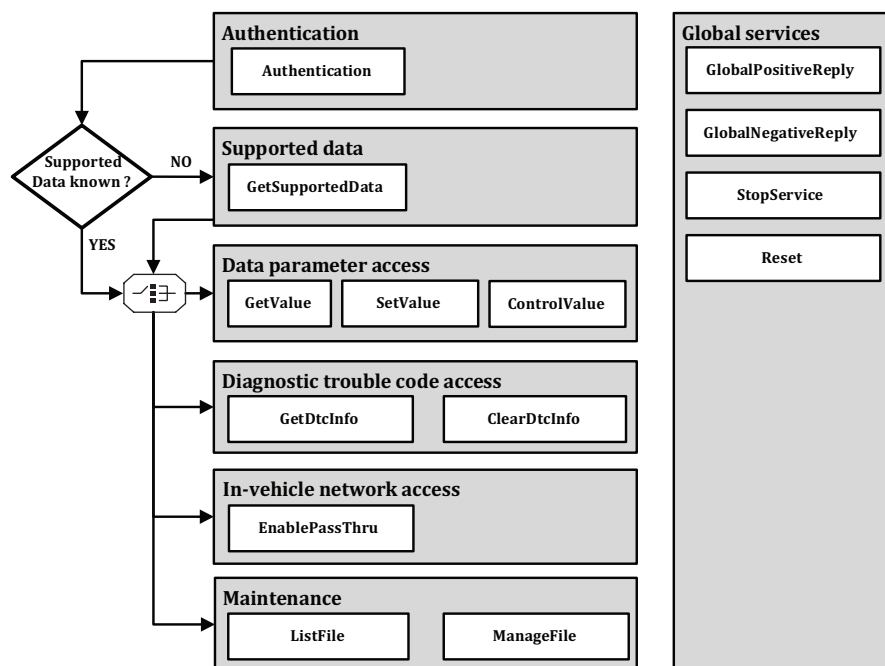


Figure 8 — UGP service clusters and associated services

9 UGP service cluster 1 — Global services

9.1 GlobalPositiveReply

Table 3 defines the global positive response message GlobalPositiveReply. List attributes are marked with {} in the attributes part of the message definition.

Table 3 — Definition of the message 'GlobalPositiveReply'

Msg	GlobalPositiveReply	Replies services requests without own reply message.	
Attributes	Name	Description	Cvt
	— no attributes —		
ASN.1	GlobalPositiveReply ::= SEQUENCE { ... }		

9.2 GlobalNegativeReply

Table 4 defines the negative response message GlobalNegativeReply for all services. It contains a list of VErrorValue elements. A VErrorValue consists of the mandatory attribute errorId as reference to an error. The number and content of the attributes rvId depend on the error. Examples for a GlobalNegativeReply are defined in the respective service.

Table 4 — Definition of the message 'GlobalNegativeReply'

Msg	GlobalNegativeReply	Possible reply on all ...Calls without results in the reply.	
Attributes	Name	Description	Cvt
	error {}	List of error values containing following sub attributes:	M
	— errorId	— Unique error identifier, references to an error	M

Table 4 (continued)

	— attribute	— List of data parameter values (see A.11)	0
ASN.1	GlobalNegativeReply ::= SEQUENCE { error SEQUENCE OF VLErrorValue, ... }		

9.3 StopService

9.3.1 Service description

The StopService is used to request the termination of a previously defined asynchronous service.

9.3.2 Message 'StopServiceCall'

[Table 5](#) defines the request message StopServiceCall using the callSequenceNumber of a previously defined asynchronous service, i.e. a service requested by a call with attribute refreshInterval greater than 0. There is no relation between the callSequenceNumber of the StopServiceCall message and the attribute.

Table 5 — Definition of the message 'StopServiceCall'

Msg	StopServiceCall	Request a service termination.	
Attrib.	Name	Description	Cvt
	callSequenceNumber	The sequence number of the asynchronous service to stop.	M
ASN.1	StopServiceCall ::= SEQUENCE { callSequenceNumber UNUM16, ... }		

9.3.3 Positive reply

If the requested service can be stopped, a GlobalPositiveReply (see [9.1](#)) is sent.

9.3.4 Error handling

If the StopService fails with an invalid callSequenceNumber or a not stoppable service, a GlobalNegativeReply (see [9.2](#)) is returned.

9.3.5 Example

An example for the StopService is included in [Table 6](#). The UGP Client requests the termination of a service with stopServiceCall by defining the callSequenceNumber 192. The V-ITS-SG replies with a positive stopServiceReply with no additional information.

Table 6 — Example for service StopService

ASN.1	<pre>stopServiceCall UGPMessage ::= { callSequenceNumber 496, choiceUGP stopServiceCall: { callSequenceNumber 192 } } stopServiceReply UGPMessage ::= { callSequenceNumber 496, choiceUGP globalPositiveReply: { } }</pre>
--------------	--

9.4 Reset

9.4.1 Service description

This `Reset` service is used to reboot or to shut down the V-ITS-SG.

9.4.2 Message 'ResetCall'

[Table 7](#) defines the request message `ResetCall`.

Table 7 — Definition of the message 'ResetCall'

Msg	ResetCall	Request a key off and on reset.	
Attributes	Name	Description	Cvt
	resetType	Type of the reset: reboot, shutdown	M
ASN.1	<pre>ResetCall ::= SEQUENCE { ... }</pre>		

9.4.3 Positive reply

A positive reply is handled by a `GlobalPositiveReply` (see [9.1](#)).

9.4.4 Error handling

If the key off and on reset does not work, a `GlobalNegativeReply` (see [9.2](#)) is returned.

9.4.5 Example

An example for the `CodeReference` is included in [Table 8](#). The UGP Client requests the reset with `resetCall`. The V-ITS-SG replies with a positive `resetReply` with no additional information.

Table 8 — Example for service Reset

ASN.1	<pre>resetCall UGPMessage ::= { callSequenceNumber 480, choiceUGP resetCall: { resetType reboot } } resetReply UGPMessage ::= { callSequenceNumber 480, choiceUGP globalPositiveReply: { } }</pre>
--------------	--