Ministry of Infrastructure
and Water Management

HGV Charge TSP

Message description data exchange RDW

Appendix to Service Description

Reading guide

This document, Message Description Data Exchange RDW, describes the message specifications provided to the Toll Service Provider (TSP) for the purpose of data exchange between TSP and Toll Charger (TC). This document is aimed to provide information to the Main Service Provider (MSP) as well as European Electronic Toll System (EETS) Providers. In this document both MSP and EETS provider will be referred to as TSP, unless a clear distinction is necessary.

Data exchange between TSP and TC will be needed at several points in the Heavy Goods Vehicle (HGV) Charge chain. Message specifications for the data exchanges are defined in the standards EN 16968:2024 and ISO 12855:2022. The EN 16986:2024 standard defines a coherent set of transactions, triggers, conditions, data elements, transfer mechanisms and support functions for an interoperable data exchange between the central systems of TC and TSP, based on ISO 12855:2022.

The EN 16986:2024 defines different profiles with specific usage for certain transactions. For the Dutch HGV Charge the SectionAutonomous profile is applicable, since tolling is executed by means of autonomous positioning recognition (GNSS) for travelling in identified areas (road sections), with the TSP playing the dominant role.

This document contains a detailed specification of each message. For more general information on the usage of each operation, please refer to document 'HGV Charge – TSP – Service description data exchange RDW.'

Although all data exchanges in this document will technically be transferred through InfoExchange messages, in this document they will be referred to by its content. E.g. an InfoExchange message containing a BillingDetailsADU will often be referred to as a Billing Details message. This is done for reading purposes in order to have a clear distinction between the different messages in this document.

This document reflects the currect state of the message description and is intended to inform the reader into detail on specifications and processing for each interaction. It is however possible that changes to this specification will be made during the realisation phase of HGV Charge project. In the event that there is a discrepancy between this document and the XSD files, the XSD files are leading and will take precedence over this document.

Change history

Version	Paragraph	Description
17-Jan-2024	-	Initial version
06-Mar-2024	2.2.1, 3	Definition of 'Toll trip' is further specified
27-Mar-2024	2.10.2	Data item listOfCccContainers is removed from Toll Declaration message
25-Apr-2024	2.3.1	Maximum length of 32 characters is added for data item paymentReference
	2.3.3	Validation for paymentReference length (PAAdu.F7) is removed. Changes made to validation PARec.F4. Validation PARec.F5 is added.
	2.4.1	Data item entryValidityEnd is added. Value 'whitelistIncremental (12) is added for data item exceptionListType. Data item vehicleTechnicalPermissibleMaxLadenMass is removed (simpleType vehicleTechnicalPermissibleMaxLadenMasses must be used)
	2.4.3	Validations for entryValidityStart and entryValidityEnd are added
	2.5.1	Data item entryValidityEnd is added to Exception List (blacklist) message
	2.5.3	Minor updates, validations for entryValidityEnd are added
	2.8.1	Data item tollContextPartitionId is removed (twice, simpleTypes tollContextPartitions and applicablePartitions must be used)
	2.9.3	Validations for Trust Objects are added
	2.11.1	Data item userParameterRequest is removed (simpleType listOf-ParametersRequested must be used)
	3	Definition of a toll trip is extended
23-May-2024		New layout and textual corrections
18-Oct-2024	-	Textual corrections
	2.4.1	Data item vehicleTechnicalPermissibleMaxLadenMasses is replaced by data item vehicleTrainMaximumWeights
	2.4.3	Change to validation WLRec.F7
	2.4.5	Added clarification regarding TIMEA_MIN for 'whitelistFull'
	2.5.3	Change to validation BLRec.F1
	2.5.5	Added clarification regarding TIMEA_MIN for 'blacklistFull'
	2.9.8	Added specific requirements regarding supported certificate
11-Jun-2025	1.7	Textual Corrections
	1.9	New paragraph 'Toll Charger identification'
	2.1 through 2.11	Corrections for alignment with XSD files; corrections and additions to Validations and errors
	2.7	Data exchange "Anonymous movement data" has been removed
	2.10.6, 2.11.6	More information added for TIME1_MAX

Table of content

1 General requirements for data exchange

This chapter describes the general functionality that is necessary for implementing a data exchange between TSP and TC. Subjects such as applicable norms, data type restrictions, basis protocol, error handling and general data items of an InfoExchange message.

1.1 General

TC and TSP use the InfoExchange for requests and confirmations. All data transmissions shall comply with the defined InfoExchange message specification in section 1.6.

The confirmation messages of ISO 12855 (e.g. InfoExchange with AckADU) are used to confirm or refuse the contents and functional correctness of the data at the application level.

A protocol for the transmission and acknowledgment of the individual messages will be provided by the underlying transport layer. The transport layer also provides functionality to ensure data confidentiality (encryption), integrity and authenticity (signature) of the messages.

Xmldsig signature checking is added and used by RDW instead of the standard ISO signing.

1.2 Norms

The interface for data exchange must meet the following norms:

- W3C WS-* norm
- ISO 12855:2022 and EN 16986:2024
- X509 v3 certificates
- "XML Signature Syntax and Processing Version 2.0" , www.w3.org/2000/09/xmldsig
- "Canonicalization algorithm" , www.w3.org/2001/10/xml-exc-c14n#
- "Signature algorithm", www.w3.org/2001/04/xmldsig-more#rsa-sha256
- "Enveloped signature type", www.w3.org/2000/09/xmldsig#enveloped-signature
- "Digest (hash) algorithm", www.w3.org/2001/04/xmlenc#sha256

1.3 Message definition

The tables in this section define the use of InfoExchange with the associated ADUs in acccordance with ISO 12855. The semantics of the data elements are applicable as defined in ISO 12855 and are not repeated in this document. The restrictions and transaction definitions of EN 16986 are complied with whenever possible.

The column "Seq" in the tables describes the sequence (or order) of data items in a message.

The column "Qty" provides information about the frequency of attributes. In addition, attributes according to the mentioned standards that are not used in the HGV Charge context are not in this document. In summary, this means:

- 0 = optional data item can be transferred, but is not mandatory
- 1 = data item shall be transmitted, exactly one data item shall be transmitted
- 0..n = none or at most n data items can be transferred,
- 1..n = at least 1 and at most n data items can be transferred.

The column "Lvl" in the tables describes the level of data items within the hierarchy of a message.

The column "Value" in the tables contains if necessary:

- The detailed value range of the type attributes. The value range or assignment of values are given in decimal numeral system unless otherwise explicitly stated.
- Additional restrictions of the type attributes.
- Deviating definitions of the attributes compared to the standard ISO 12855, EN ISO 17575-1 and EN 16986 are denoted by Top-Up.
- Additional descriptions of the type attributes.

The rows in the tables in section 2 are highlighted using the following rules and colour codes to provide a better overview and give some additional information:

- attributes to be transmitted are highlighted by noting these against white background
- attributes according to standards to be submitted but not used are highlighted against orange background
- and attributes and/or attribute values that are Top-Up to standards are highlighted against green background

1.4 General requirements

During a toll trip the userId, the combination of pan, obeId and licencePlateNumber shall not change.

1.5 Data type restrictions

For data elements that have not been otherwise restricted in terms of scope or format, the following restrictions apply:

- In the ASN.1 definition unlimited INTEGER shall not exceed the maximum value 2^63-1. For some data elements a specific maximum value is specified. The ASN.1 INTEGER type is converted into one of several XSD built-in types depending on value range constraints on the integer type definition.

  If the integer has a value range constraint that allows a more restrictive XSD type to be used, then that type will be used. For example, if a range of 0 to 255 (inclusive) is specified, an XSD unsignedByte would be used because it maps exactly to this range. The following table shows the range values for each of the INTEGER type mappings:

Lower bound	Upper bound	XSD data type	Negative values
-128	127	byte	Yes
0	255	unsignedByte or Int1Unsigned	No
-32768	32767	short	Yes
0	65535	unsignedShort or Int2Unsigned	No
-2147483648	2147483647	Integer or int	Yes
0	4294967295	unsignedInt	No
-9223372036854775808	9223372036854775807	long	Yes
0	18446744073709551615	unsignedLong	No

- For some data elements, there may be a more strict limitation of the value range. The specification of these limitations can be found in the XSD files.

- In the ASN.1 definition of non-limited types “UTF8String” or “OCTET STRING” shall not include more than 1024 characters or octets. For some attributes a longer value is however allowed; if applicable, this will be mentioned for each specific attribute in this document.

- All time data elements without a time base defined shall be in coordinated universal time (UTC).

- Data elements of the type GeneralizedTime (DateTime) shall have, according to ISO/IEC 8824-1, section 46.3, the format b) with a resolution of one second and shall be in coordinated universal time.

1.6 XSD files

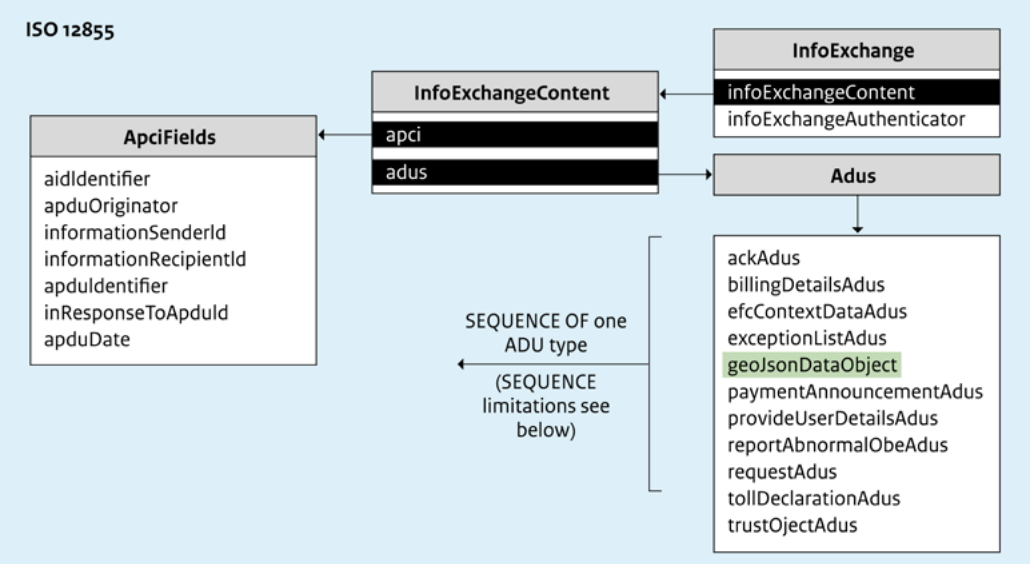
The TC will provide a complete set of XSD files to TSP. This set of XSD files starts with the ISO 12855 module and includes

- all required definitions from EN 16986,
- all Top-Up definitions from the current document,
- and all required sub modules. The XSD files have been generated from of the following ASN.1 files:
  - ISO12855(2021)EfcInfoExchangev5.asn
  - ISO17573-3(2023)EfcDataDictionaryV1.3.asn
  - ISO12813(2022)EfcCccV5.asn
  - ISO13141(2023)EfcLac2.1.asn.unused
  - ISO13141(2023)EfcLacV3.asn
  - ISO14816(2015)AVIAEINumberingAndDataStructures.asn
  - ISO14906(2023)GenericV10.asn
  - ISO14906(2023)ApplicationV9.asn
  - a custom ASN.1 file containing the added elements of EN 16986 and the Top-up elements

The provided XSD files will contain enumeration and value range restrictions defined in the current document. In the event that there is a discrepancy between this document and the XSD files, the XSD files are leading and will take precedence over this document.

1.7 Basis protocol

The basic protocol is based on the InfoExchange from ISO 12855 and shown below. For Adus only the variants required for the Dutch HGV Charge interface are listed.



An InfoExchange shall contain a SEQUENCE OF with only one single ADU; with the exception for messages:

- billingDetails
- provideUserDetails
- tollDeclaration

For more details about the geoJsonDataOject, please refer to: 2.10.2 Specification “TollDeclarationADU” and 2.10.7.1 Specification “geojson” (as part of a Toll declaration).

1.8 Message structure – InfoExchange data items

All exchanged messages are transmitted with the InfoExchange as defined below.

Seq	Qty	Lvl	Data item	Type	Value
1	1	0	infoExchange		
2	1	1	infoExchangeContent		
3	1	2	apci		
4	1	3	aidIdentifier	unsignedByte	Value: cen16986x2022 (17) *Date "2022" potentially to be adjusted*
5	1	3	apduOriginator		
6	1	4	countryCode	Bitstring 10	
7	1	4	providerIdentifier	unsignedShort	
8	1	3	informationSenderID		
9	1	4	countryCode	Bitstring 10	
10	1	4	providerIdentifier	unsignedShort	
11	1	3	informationRecipientID		
12	1	4	countryCode	Bitstring 10	
13	1	4	providerIdentifier	unsignedShort	
14	1	3	apduIdentifier	unsignedLong	
15	0	3	inResponseToApduId		
16	1	4	apduOriginator		
17	1	5	countryCode	Bitstring 10	
18	1	5	providerIdentifier	unsignedShort	
19	1	4	apduIdentifier	unsignedLong	
20	1	3	apduDate	DateTime	
21	1..n	2	adus *Containing the ADU content*		
			This data item will be following the 'adus' data item:		
+1	0	1	Signature	*Refer to XSD*	

1.9 Toll Charger identification

The CS1 issuer identifier number of the RDW for HGV Charge must be used to identify the Toll Charger in all data exchanges described in this document. The value for the Toll Charger for HGV Charge in the Netherlands, represented by RDW, can be found in the table below.

Data item	Value	Meaning/Description
Countrycode	0011001001	*NL - The Netherlands*
providerIdentifier	6	*Dienst Wegverkeer (RDW) - Vrachtwagenheffing*

1.10 Error handling

The correct syntax of the whole message will be checked by the transport layer. Possible errors are described for each message.

In case of signing and syntactical errors the APDU will not be processed and a technical error response shall be sent.

In case of content and logical errors of the application protocol control information (apci) part of the APDU, an AckADU according to section 2.1.1 with apduAckCode (without an issue) listed shall be sent.

Technical checks
Messages are offered on the endpoint of the TC to be further checked and processed from there. In case of a technical error, a synchronous HTTP error is returned.

Functional messages from the system of the TC are fed back asynchronously via the ACK message. See: 2.1.1 Specification 'AckADU'.

The technical checks in the process are handled first and will be returned synchronous as a HTTP response:
- HTTP 200 for successful received.
- HTTP 400 for e.g. problems with the signing, authentication or XSD-scheme.
- HTTP 500 for internal server errors.

Some of the errors will be returned in the format of the ACK message.

The following technical checks will be done before accepting an incoming message.

Check	In case of an error
Checking the client certificate on expiration	HTTP400
Checking the client certificate on revocation	HTTP400
Checking the client certificate on trusted root certificate	HTTP400
Checking if the client certificate belongs to the registered service provider	*HTTP400*
Checking the signing certificate on expiration	ACK-response
Checking the signing certificate on revocation	ACK-response
Checking the signing certificate on trusted root certificate	ACK-response
Checking if the signing certificate belongs to the registered service provider	HTTP400
Checking the signature and hash of the message *Please note: xmldsig signature checking is used instead of the standard ISO signing.*	HTTP400
XSD-scheme validation	HTTP400
Check on "unique message id" (to prevent duplication)	HTTP400

2 Interactions and messages

This chapter contains a detailed description of messages for each interaction. Validation rules, timing parameters and specific requirements for each interaction are also described.

2.1 General

The messages AckADU and RequestADU are used in several interactions.

An AckADU is generally used as a response to an ADU transmission.

A RequestADU is used by TC to ask TSP to provide certain information in an ADU-message (TollDeclarationADU, ProvideUserDetailsADU).

2.1.1 Specification 'AckADU'

Seq	Qty	Lvl	Data item	Impl.	Value
1-20			See: 1.8 Message structure – InfoExchange data items		
21	1	2	ackAdus		
22	1..n	3	ackAdu		
23	1	4	apduIdentifier	unsignedLong	
24	1	4	apduAckCode	unsignedByte	
25	0	4	apduAckText	String 1024	
26	0..n	4	issues		
27	1	5	sequence		
28	0	6	issueAduIdentifier	unsignedLong	
29	0	6	issueLocation	String 512	
30	0	6	issueContent	String 1024	
31	1	6	issueCode	unsignedShort	
32	0	6	issueText	String 1024	
33	0	4	actionCode	unsignedByte	Send (0)

2.1.1.1 *Specific requirements*

There are some specific requirements for the exchange of an AckADU message.

- An InfoExchange shall contain only one AckADU

2.1.2 Specification 'RequestADU'

Seq	Qty	Lvl	Data item	Impl.	Value
1-20			See: 1.8 Message structure – InfoExchange data items		
21	1	2	requestAdus		
22	1..n	3	RequestAdu		
23	1	4	*<requestType>**		
24	1..n	4	*<requestParameters>**		
25	1	4	actionCode	unsignedByte	Send (0)

* *Refer to sections 2.10 and 2.11 for more details.*

2.1.2.1 *Specific requirements*

There are some specific requirements for the exchange of an RequestADU message.

- An InfoExchange shall contain only one RequestADU

2.2 Data Exchange “Billing details”

TSP will send a Billing details message
- for each toll trip that has ended; or
- for a ‘zero km-notification’ (MSP-only)

2.2.1 Specification ‘BillingDetailsADU’

Seq	Qty	Lvl	Data item	Impl.	Value
1-20			See: 1.8 Message structure – InfoExchange data items		
21	1	2	billingDetailsAdus		
22	1..n	3	BillingDetailsAdu		
23	1	4	aduIdentifier	unsignedLong	
24	1	4	tollContext		
25	1	5	countryCode	Bitstring 10	
26	1	5	providerIdentifier	unsignedShort	
27	1	4	userId		
28	0	5	pan	hexBinary 10	
29	0	5	licencePlateNumber		
30	1	6	countryCode	Bitstring 10	
31	1	6	alphabetIndicator	Token	Value: latinAlphabetNo1
32	1	6	licencePlateNumber	hexBinary	
33	0	5	obeId		
34	1	6	manufacturerId	unsignedShort	
35	1	6	equipmentObuId	hexBinary	
36	1	4	period		
37	1	5	beginOfPeriod */**	DateTime	
38	1	5	endOfPeriod */**	DateTime	
39	1	4	billingDetailsAmount		
40	1	5	paymentFeeAmount	long	
41	1	5	paymentFeeUnit	hexBinary 2	Value: ‘2978’ (euro-cents)
42	1	4	usage		
43	1	5	usageList		
44	1..n	6	UsageList		
45	1	7	usageListEntry		
46	1	8	forSectionedRoads		
47	1	9	howManyTimes	long	Value: 1

Seq	Qty	Lvl	Data item	Impl.	Value
48	0	9	appliedTariffTableVersion	hexBinary	
49	1	9	appliedLocalVehicleClass		
50	0	10	appliedLocalVehicleClass Id	unsignedShort	
51	0	10	appliedLocalVehicleClass Description	String	
52	1	9	appliedTimeClass		
53	0	10	appliedTimeClassId	Int2Unsigned	Value: 1
54	0	10	appliedTimeClass Description		
55	1	9	appliedUserClass	string	
56	0	10	appliedUserClassId	Int1Unsigned	Value: 1
57	0	10	appliedUserClass Description	string	
58	0	9	usageDistance		
59	1	10	distanceValue	unsignedInt	
60	1	10	distanceUnit	unsignedByte	Value: 'meters' or '2'
61	1	4	refTollDeclaration *Unique reference to a TollDeclaration*		
62	1	5	CHOICE *Must contain a refTollDeclarationId*		
63	1..n	6	refTollDeclarationId		
64	1	6	issuerId		
65	1	7	countryCode	Bitstring 10	
66	1	7	providerIdentifier	unsignedShort	
67	1	6	declarationId	unsignedLong	
68	0	4	actionCode	unsignedByte	Value: Send (0)

* *For a chargeable toll trip the beginOfPeriod must contain the date and time of entering the chargeable network (entering the first chargeable road section); the endOfPeriod must contain the date and time of leaving the chargeable network (meaning leaving the final chargeable road section of that trip).*

** *For a 'zero km-notification' (MSP only) the beginOfPeriod must be 00:00:00 of a calender day; the endOfPeriod must be 23:59:59 for the same calender day.*

2.2.2 Response

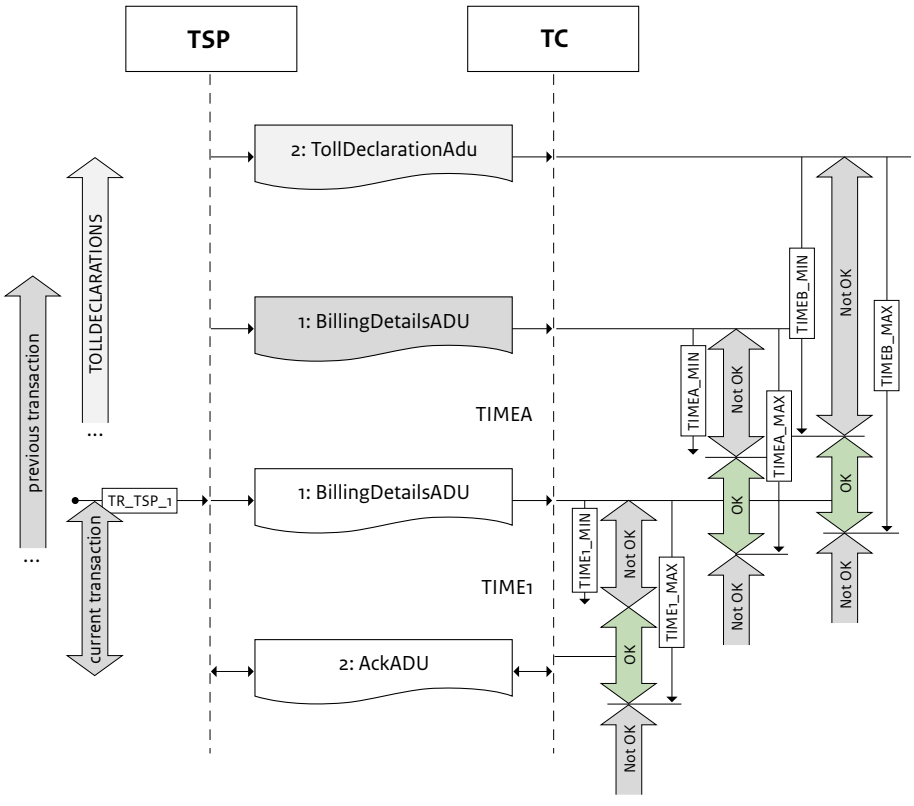
An AckADU is sent by TC in response to an InfoExchange message containing a BillingDetailsADU that is sent by TSP. See: section 2.1.1 AckADU.

2.2.3 Validation and errors

A billingDetailsADU is functionally validated. The following errors will be returned, if validations are not successful.

#	Check	Validation	Error (if not successful)
1 BDRec.F1	userId	Is there an active whitelist entry for this service user (pan, LPN and obeId)?	Reason code 704 - User Id incorrect
2 BDRec.F2	period	- beginOfPeriod must be before endOfPeriod and - endOfPeriod may not be in the future	Reason code 705 - Period is not valid
3 BDRec.F3	PaymentFeeUnit	PaymentFeeUnit must contain value for 'Eurocents' (2978)	Reason code 706 - PaymentFee-Unit is not valid
4 BDRec.F5	DistanceUnit	DistanceUnit contains value for 'meters' or '2'	Reason code 3000 - DistanceUnit is not valid
5 BDRec.F7	duplicate billingDetails	check if aduIdentifier of billingDetails is either new or duplicate	Reason code 3000 - duplicate billingDetails id
6 BDRec.F8	actionCode	actionCode must contain value 'send' (0)	Reason code 3010 - actionCode-NotSupported
7 BDRec.F9	usageList	Must contain exactly one element	Reason code 3000 - Invalid number of usage list entries
8 BDRec.F10	tollContext	Must be equal to the value of the originator.providerId (apci) of the efcContextData (the Toll Charger)	Reason code 702 - Toll Charger ID is not valid
9 BDRec.F11	refTollDeclarationId	refTollDeclarationId is present and declarationId is not empty	Reason code 3000 – refTollDecla-rationId is missing or empty
10 BDRec.F12	TollDeclarationId - issuerId	Provider (issuerIdentifier and country-Code) of Tolldeclaration must match values of apduOriginator	Reason code 3000 – Provider of TollDeclarationId is not correct

2.2.4 Sequence diagram



Sequence diagram of the BILLINGDETAILS_TSP transaction from EN 16986:2024.

Note: *In the HGV Charge implementation of the BILLINGDETAILS_TSP transaction a BillingDetailsADU is not preceded by a TollDeclarationAdu. The TollDeclarationADU as shown in this sequence diagram must therefore be ignored. Timing parameters TIMEB_MIN and TIMEB_MAX do not apply.*

2.2.5 Timing parameters

The table below contains the relevant timing parameters for BILLINGDETAILS_TSP for HGV Charge.

Timing	Value	More information
TIMEA_MIN	0	Billing details messages can be sent in parallel. Therefore TIMEA_MIN is defined as 0
TIME1_MIN	0	
TIME1_MAX	24 hours	

TIMEB_MIN and TIMEB_MAX are not applicable for the Dutch HGV Charge implementation.

2.2.6 Specific requirements

There are some specific requirements for the exchange of a Billing details message.

- An InfoExchange shall contain one or more BillingDetailsADU's.
- A BillingDetailsADU contains only the declaration of a single toll trip (or a 'zero km-notication') for a single vehicle.
- Information such as movement data is not transferred, but is kept by TSP and is available to be requested by TC in a TollDeclarationADU. For the purpose of such a request, the refTollDeclaration in the BillingDetailsADU will contain an unique reference to the movement data. See also: TollDeclarationADU.
- The transmission of a toll trip declaration by TSP is considered successful only after it has been acknowledged by TC with an AckADU containing an 'apduOk' (where no single issue/error has occurred for that specific ADU).

2.3 Data exchange "Payment announcement"

In order to transfer the owed HGV levies to the TC, the TSP needs to send a Payment announcement message. This message is used to announce payments that the TSP will be making, including a reference to the billing details for which the payment will be made.

2.3.1 Specification 'PaymentAnnouncementADU'

Seq	Qty	Lvl	Data item	Impl.	Value
1-20			See: 1.8 Message structure – InfoExchange data items		
21	1	2	paymentAnnouncementAdus		
22	1..n	3	paymentAnnouncementAdu		
23	1	4	aduIdentifier	long	
24	1	4	dueDate	DateTime	
25	0	4	totalAmount		
26	0	5	paymentFeeAmount	long	

Seq	Qty	Lvl	Data item	Impl.	Value
27	0	5	paymentFeeUnit	hexBinary 2	
28	1	4	paymentStatus	unsignedByte	Value: notYetDue (2)
29	1..n	4	referenceDetailsList		
30	1..n	5	SEQUENCE		
31	1	5	referenceDetailList		
32	1..n	6	referenceDetail		
33	1	7	referencedAduIdentifier		
34	1	8	referencedAduIdentifier	unsignedLong	
35	1	8	referencedAduType	unsignedByte	Value: '8' or 'billingDetailsAdu'
36	1	5	Amount		
37	1	6	paymentFeeAmount	long	
38	1	6	paymentFeeUnit	hexBinary	Value: '2978' (euro-cents)
39	1	4	paymentReference	string 32	
40	1	4	actionCode	unsignedByte	Value: Send (0)

2.3.2 Response

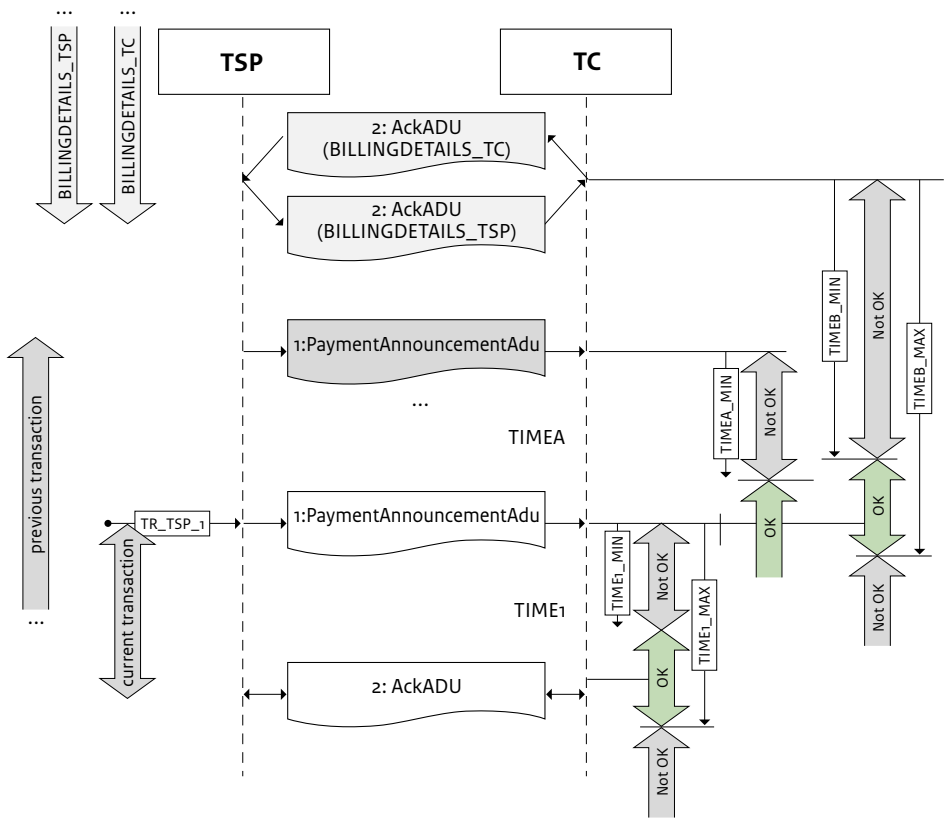
An AckADU is sent by TC in response to an InfoExchange message containing a PaymentAnnouncementADU that is sent by TSP. See: section 2.1.1 AckADU.

2.3.3 Validation and errors

A paymentAnnouncementADU is functionally validated. The following errors will be returned, if validations are not successful.

#	Check	Validation	Error (if not successful)
1 PAAdu.F1	referenceDetailList	referenceDetailList must be present	Reason code 3000 - Reference detail list missing
2 PAAdu.F3	paymentFeeUnit	PaymentFeeUnit must contain value for 'Eurocents' (2978)	Reason code 3000 - Invalid Payment fee unit
3 PAAdu.F4	paymentStatus	paymentStatus must contain value 'notYetDue' (2)	Reason code 3000 - Invalid Payment status
4 PAAdu.F5	paymentReference	paymentReference must have a value	Reason code 3000 - Payment reference missing
5 PAAdu.F6	actionCode	actionCode must contain value 'send' (0)	Reason code 3010 - Action code not supported
7 PARec.F1	referencedAduType	referencedAduType must contain value 'billingDetailsAdu' (8)	Reason code 3000 - Invalid referenced ADU type
8 PARec.F2	referenced-AduIdentifier	referencedAduIdentifier must have been exchanged successfully in a BillingDetailsADU	Reason code 3000 - Unknown referenced ADU identifier
9 PARec.F3	aduIdentifier	This aduIdentifier has not yet been successfully exchanged in a PaymentAnnouncementADU	Reason code 3000 - Duplicate paymentannouncement for billingdetail
10 PARec.F4	amount	Must be equal to amount in referenced BillingDetails	Reason code 3000 - Billing amount mismatch
11 PARec.F5	totalAmount	totalAmount must equal the sum of all referenced detail amounts	Reason code 3000 – Total amount mismatch

2.3.4 Sequence diagram



Sequence diagram of the PAYMENTANNOUNCEMENT transaction from EN 16986:2024.

2.3.5 Timing parameters

The table below contains the relevant timing parameters for PAYMENTANNOUNCEMENT for HGV Charge.

Timing	Value	More information
TIME1_MIN	0	
TIME1_MAX	24 hours	

2.3.6 Specific requirements

There are some specific requirements for the exchange of a Payment announcement message.

- An InfoExchange shall contain only one PaymentAnnouncementADU.
- Format of a paymentReference value is:
  <Issuer ID>-<Countrycode (two letters)>-“HGVC”-<ddmmyy>-<sequence number; starting at 01>
  Example for a fictitious TSP from the Netherlands with issuer ID 99 for a Payment announcement that is sent 14 February 2025: paymentReference = 99-NL-HGVC-140225-01
- Each referenced toll trip, that has been successfully exchanged through a Billing details message, must be included in a separate occurrence of SEQUENCE, in order for the TC to match referencedDetail and Amount per trip.
- The transmission of a payment announcement by TSP is considered successful only after it has been acknowledged by TC with an AckADU (no single issue/error has occured in that specific ADU).

2.4 Data exchange “Exception list (whitelist)”

After concluding a service agreement between TSP and holder (service user), the service user and details of its vehicle will be added to the whitelist of the TSP. Each day a full list is sent by TSP to TC containing all service users for which the TSP will accept responsibility. An incremental list update may also be used in addition to the full list.

Note: *The ExceptionListADU structure will be used for the exchange of whitelists and blacklists. For reading purposes whitelist and blacklist are described separately although they share the same ADU structure.*

2.4.1 Specification ‘ExceptionListADU’ (type: whitelist)

Seq	Qty	Lvl	Data item	Impl.	Value
1-20			See: 1.8 Message structure – InfoExchange data items		
21	1	2	exceptionListAdus		
22	1..n	3	ExceptionListAdu		
23	1	4	aduIdentifier	unsignedLong	
24	1	4	exceptionListVersion	unsignedLong	
25	1	4	exceptionListType	unsignedByte	Values: whiteListFull (2), whiteListIncremental (12)
26	0	4	exceptionValidityStart	DateTime	
27	1	4	exceptionListEntries		
28	0..n	5	exceptionListEntry		
29	1	6	userId		
30	0	7	pan	hexBinary 10	
31	0	7	licencePlateNumber		
32	1	8	countryCode	Bitstring 10	
33	1	8	alphabetIndicator	token	Value: latinAlphabetNo1
34	1	8	licencePlateNumber	hexBinary	
35	0	7	obeId		
36	1	8	manufacturerId	unsignedShort	
37	1	8	equipmentObuId	hexBinary	
38	1	6	reasonCode		Value: whiteListedUser (8)
39	1..n	7	exceptionListReasonType	unsignedByte	
40	0	6	entryValidityStart	DateTime	
41	0	6	entryValidityEnd	DateTime	
42	0	6	nominalVehicleParameters		
43	1	7	vehicleClasses	string	
44	0	7	co2Classes		
45	1..n	8	co2Class		
46	1	9	co2Scheme	unsignedByte	Value: eU (1)
47	1	9	co2EmissionClass	unsignedByte	
48	0	7	euroValues		

Seq	Qty	Lvl	Data item	Impl.	Value
49	1..n	8	euroValue	unsignedByte	
50	0	7	vehicleTrainMaximum-Weights	unsignedShort	Value shall be expressed in 10 kg, rounded down to the next 10 kg
51	0	6	actionRequested	unsignedByte	
52	0	4	actionCode	unsignedByte	Value: Send (0), Adjust (3)

2.4.2 Response

An AckADU is sent by TC in response to an InfoExchange message containing a ExceptionListADU that is sent by TSP. See: section 2.1.1 AckADU.

2.4.3 Validation and errors

An exceptionListADU (whitelist) is functionally validated. The following errors will be returned, if validations are not successful.

#	Check	Validation	Error (if not successful)
1 WLAdu.T1	count (exceptionList-Adu)	Must be 1	Reason code = 3000 - apduAckText 'Too many exceptionListAdus'
2 WLAdu.F1	exceptionListType	exceptionListType must contain value whiteListFull (2) or whiteList-Incremental (12)	Reason code = 401 – exceptionList-sTypeRejected
3 WLAdu.F2	exception-ValidityStart	exceptionValidityStart may not be in the future	Reason code = 405 - exceptionList-sDateAndTimeRejected: date in future not allowed
4 WLAdu.F4	actionCode	actionCode must contain Send (0) for exceptionListType = whitelist-Full and Adjust (3) for exception-ListType = whiteListIncremental	Reason code = 3010 – actionCode-NotSupported

After an ExceptionListADU (whitelist) is validated successfully each whitelist entry is validated individually.

#	Check	Validation	Error (if not successful)
1 WLRec.F1	reasonCode	Value must be equal to white-ListedUser (8)	Reason code = 404 - The reason code in the exception list has been rejected
2 WLRec.F2	countryCode	Value must be a valid country (ISO 3166-1, Alpha-2)	Reason code = 402 - exception-ListsUserIdRejected: invalid countryCode
3 WLRec.F3	vehicleLicencePlate-Number	Value of alphabetIndicator must be latinAlphabetNo1	Reason code = 402 - exception-ListsUserIdRejected: invalid alphabetIndicator
4 WLRec.F4	entryValidityStart	Value may not be in the future	Reason code = 405 - exception-ListsDateAndTime Rejected: date in future not allowed
5 WLRec.F5	entryValidityStart	Value must be between exception-ValidityStart of previous full exceptionList (type: WhitelistFull) and exceptionValidityStart of this exceptionList	Reason code = 405 – exception-ListsDateAndTime Rejected: not between current and previous list

#	Check	Validation	Error (if not successful)
6 WLRec.F6	euroValues	Value is present; value is 0 to 7	Reason code = 3000 - Invalid EuroValue
7 WLRec.F7	vehicleTrain-MaximumWeights	Value is present	Reason code = 3000 - Required element
8 WLRec.F8	co2Class	Value is present; value is 0 to 5	Reason code = 3000 - Invalid co2 class
9 WLRec.F9	vehicleClasses[1]	Value is present; must contain a valid vehicleClass	3000 - Required element 3000 – Invalid vehicle class
10 WLRec.F10	entryValidityEnd (if present)	Value may not be before entryValidityStart	405 - exceptionListsDateAndTime Rejected: entryValidityEnd less than entryValidity
11 WLRec.F11	entryValidityStart	Value may not be changed for an active entry	405 - exceptionListsDateAndTime Rejected: entryValidityEnd cannot be changed
12 WLRec.F12	entryValidityEnd (if present)	Value may not be changed for an active entry	405 - exceptionListsDateAndTime Rejected: entryValidityEnd cannot be changed
13 WLRec.F 13	entryValidityEnd (if present)	Value may not be in the future	405 - exceptionListsDateAndTime Rejected: date in the future not allowed
14 WLRec.F 14	entryValidityEnd (if present)	Value must be between exception-ValidityStart of previous full exceptionList (type: WhitelistFull) and exceptionValidityStart of this exceptionList	Reason code = 405 – exception-ListsDateAndTime Rejected: not between current and previous list
15 WLRec.F 15	VehicleLicencePlate-Number	Value may only contain letters (A-Z) and numbers (0-9); no spaces or other characters are allowed. Additionally for Germany (DE): - Letters Ä, Ö and Ü are also allowed - Exactly one space character must be present at the second, third or fourth position (following the letter or letters identifying the regional code). Additionally for Austria (AT): - Exactly one space character must be present at the second or third position (following the letter or letters identifying the regional code).	Reason code = 3000 - Invalid vehicle license plate number

If an entry validation is not successful this does not effect other entry validations as well as the general message validation.

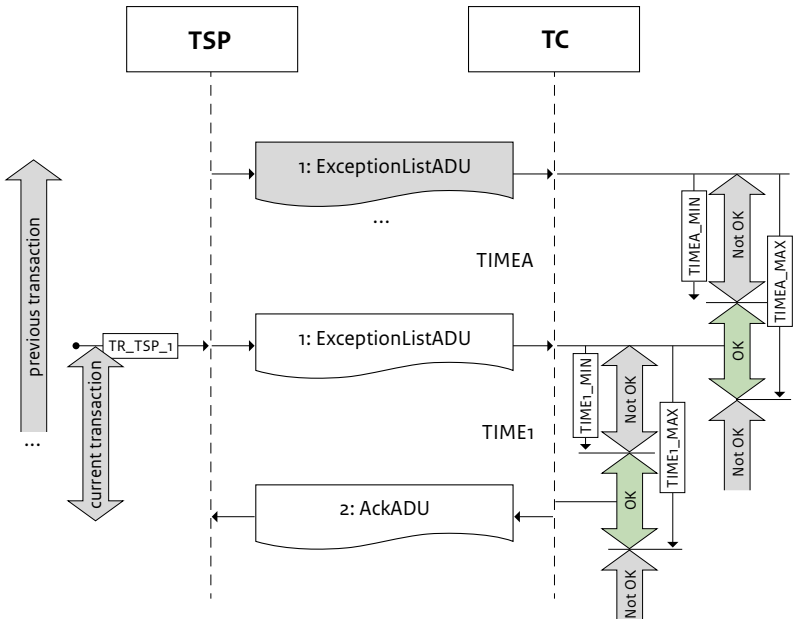
Examples

1. An ExceptionListADU contains an exceptionValidationStart in the future. An AckADU is sent containing an 'apduOk' and an issue for the reason for rejecting the entire ADU.

2. An ExceptionListADU contains 1,000 entries; the general ADU values are validated successfully, but two entries cannot be validated. 998 entries are processed succesfully. An AckADU is sent containing an 'apduOk' and an issue for each of the two rejected entries.

3. An ExceptionListADU contains 1,000 entries; the general ADU values are validated successfully, but all 1,000 entries cannot be validated. An AckADU is sent containing an 'apduOk' and 1,000 issues for the entry errors.

2.4.4 Sequence diagram



Sequence diagram of the EXCEPTIONLIST transaction from EN 16986:2024.

2.4.5 Timing parameters

The table below contains the relevant timing parameters for EXCEPTIONLIST for HGV Charge, specific per exceptionListType.

For 'whiteListFull':

Timing	Value	More information
TIMEA_MIN*	20 hours	Full WhiteLists must be transferred every 24 hours, therefore TIMEA_MIN is defined as 20 hours.
TIMEA_MAX	28 hours	Full WhiteLists must be transferred every 24 hours, therefore TIMEA_MAX is defined as 28 hours.
TIME1_MIN	0 seconds	
TIME1_MAX	24 hours	

* *Note: TIMEA_MIN is applicable for successfull exchanges. If an exchange is not successfull, this timing parameter does not apply since a retry must be performed as soon as possible.*

For 'whiteListIncremental':

Timing	Value	More information
TIMEA_MIN	0 seconds	
TIMEA_MAX	*Undefined*	TIMEA_MAX cannot be defined
TIME1_MIN	0 seconds	
TIME1_MAX	4 hours	

Time window for full whitelist transfer
The TSP must successfully exchange exactly one exceptionListAdu of exceptionListType 'whiteListFull' with the TC per day. The TSP must send the TC this exceptionListAdu during the night between 4:00 and 7:00 (AM).

2.4.6 Specific requirements

There are specific requirements for the exchange of an Exception list (whitelist) message.

- An InfoExchange shall contain only one ExceptionListADU.
- The transmission of a exception list by TSP is considered successful only after it has been acknowledged by TC with an AckADU containing an 'apduOk'.
- One or more individual whitelist entries can be rejected. One or more issues shall be provided to TSP in the AckADU per rejected entry. All other (approved) entries have been processed succesfully.

2.5 Data exchange "Exception list (blacklist)"

A blacklist contains vehicles (service users) for which the TSP will temporarily not accept responsibility. This list is used for reporting an exception for an active service user, for example, reporting a faulty OBU or a suspension following a non-payment. Each day a full list is sent by TSP to TC containing all active exceptions. An incremental list update may also be used in addition to the full list.

Note: *The ExceptionListADU structure will be used for the exchange of whitelists and blacklists. For reading purposes whitelist and blacklist are described separately although they share the same ADU structure.*

2.5.1 Specification 'ExceptionListADU' (type: blacklist)

Seq	Qty	Lvl	Data item	Impl.	Value
1-20			See: 1.8 Message structure – InfoExchange data items		
21	1	2	exceptionListAdus		
22	1..n	3	ExceptionListAdu		
23	1	4	aduIdentifier	unsignedLong	
24	1	4	exceptionListVersion	unsignedLong	
25	1	4	exceptionListType	unsignedByte	Value: blackListFull (1), blackListIncremental (11)
26	0	4	exceptionValidityStart	DateTime	
27	1	4	exceptionListEntries		

Seq	Qty	Lvl	Data item	Impl.	Value
28	0..n	5	exceptionListEntry		
29	1	6	userId		
30	0	7	pan	hexBinary 10	
31	0	7	licencePlateNumber		
32	1	8	countryCode	Bitstring 10	
33	1	8	alphabetIndicator	token	Value: latinAlphabetNo1
34	1	8	licencePlateNumber	hexBinary	
35	0	7	obeId		
36	1	8	manufacturerId	unsignedShort	
37	1	8	equipmentObuId	hexBinary	
38	1	6	reasonCode	unsignedByte	Value: 1-7 or 13-14 or 18-19 or 30-31
39	0	6	entryValidityStart	DateTime	
40	0	6	entryValidityEnd	DateTime	
41	0	6	actionRequested	unsignedByte	
42	0	4	actionCode	unsignedByte	Value: Send (0), Adjust (3)

2.5.2 Response

An AckADU is sent by TC in response to an InfoExchange message containing a ExceptionListADU that is sent by TSP. See: section 2.1.1 AckADU.

2.5.3 Validation and errors

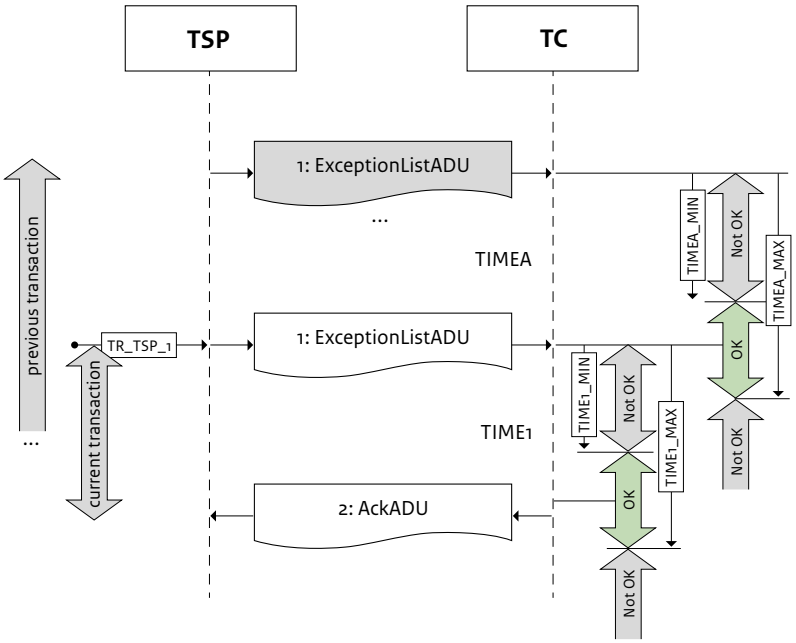
An exceptionListADU is functionally validated. If these validations are not successful, an AckADU with an 'ApduOk' wil be returned containing one or more of the following errors.

#	Check	Validation	Error (if not successful)
1 BLAdu.T1	count (exceptionList-Adu)	Must be 1	apduAckText 'Too many exception-ListAdus'
2 BLAdu.F1	exceptionListType	exceptionListType value must be blackListFull (1) or blackList-Incremental (11)	Reason code = 401 – exceptionList-sTypeRejected
3 BLAdu.F3	exceptionValidity-Start	exceptionValidityStart may not be in the future	Reason code = 405 - exceptionList-sDateAndTimeRejected: date in future not allowed
4 BLAdu.F4	actionCode	actionCode must Send (0) for exceptionListType = blacklistFull and Adjust (3) for exceptionList-Type blackListIncremental	Reason code = 3010 – actionCode-NotSupported

If the exceptionListADU is validated succesfully, each individual list entry will be validated separately. The AckADU, with an 'ApduOk', can therefore contain issues for one or more individual list entries.

#	Check	Validation	Issue (if not successful)
1 BLRec.F1	reasonCode	Value must be equal to one of 1-7 or 13-14 or 18-19 or 30-31. *Other values are not allowed*	404 - The reason code in the exception list has been rejected
2 BLRec.F2	countryCode	Value must be a valid country (ISO 3166-1, Alpha-2)	Reason code = 402 - exceptionListsUserIdRejected: invalid countryCode
3 BLRec.F3	vehicleLicencePlateNumber	Value of alphabetIndicator must be latinAlphabetNo1	402 - exceptionListsUserIdRejected: invalid alphabetIndicator
4 BLRec.F4	entryValidityStart	Value is present. Value may not be in the future	3000 - Required element 405 - exceptionListsDateAndTimeRejected: date in future not allowed
5 BLRec.F5	entryValidityStart	Value must be between exceptionValidityStart of previous full exceptionList (type: blacklistFull) and exceptionValidityStart of this exceptionList	405 - exceptionListsDateAndTimeRejected: not between current en previous list
6 BLRec.F6	entryValidityEnd (if present)	Value may not be before entryValidityStart	405 - exceptionListsDateAndTimeRejected: date before start not allowed
7 BLRec.F7	entryValidityStart	entryValidityStart for an active entry has been changed since last exceptionList	405 - exceptionListsDateAndTimeRejected: entryValidityStart cannot be changed
8 BLRec.F8	entryValidityEnd (if present)	entryValidityStart for an ended entry has been changed since last exceptionList	405 - exceptionListsDateAndTimeRejected: entryValidityEnd cannot be changed
9 BLRec.F9	entryValidityEnd (if present)	Value may not be in the future	405 - exceptionListsDateAndTimeRejected: date in future not allowed
10 BLRec.F10	entryValidityEnd (if present)	Value must be between exceptionValidityStart of previous full exceptionList (type: blacklistFull) and exceptionValidityStart of this exceptionList	405 - exceptionListsDateAnd-TimeRejected: not between current en previous list

2.5.4 Sequence diagram



Sequence diagram of the EXCEPTIONLIST transaction from EN 16986:2024.

2.5.5 Timing parameters

The table below contains the relevant timing parameters for EXCEPTIONLISTADU for HGV Charge, specific per exceptionListType.

For 'blackListFull':

Timing	Value	More information
TIMEA_MIN*	20 hours	
TIMEA_MAX	28 hours	
TIME1_MIN	0 seconds	
TIME1_MAX	24 hours	

* *Note: TIMEA_MIN is applicable for successfull exchanges. If an exchange is not successfull, this timing parameter does not apply since a retry must be performed as soon as possible.*

For 'blackListIncremental':

Timing	Value	More information
TIMEA_MIN	0 seconds	
TIMEA_MAX	*Undefined*	TIMEA_MAX cannot be defined
TIME1_MIN	0 seconds	
TIME1_MAX	4 hours	

Time window for full blacklist transfer
The TSP must successfully exchange exactly one exceptionListAdu of exceptionListType 'blackListFull' with the TC per day. The TSP must send the TC this exceptionListAdu during the night between 4:00 and 7:00 (AM).

2.5.6 Specific requirements

There are some specific requirements for the exchange of an Exception list (blackListFull or blackListIncremental) message.

- An InfoExchange shall contain only one ExceptionListADU
- The transmission of an exception list by TSP is considered successful only after it has been acknowledged by TC with an AckADU (with an 'ApduOk'). Issues in individual entries will be allowed.
- One or more single blacklist entries can be rejected. One or more issues shall be provided to TSP in the AckADU per rejected entry. All other (approved) entries have been processed succesfully.

2.6 Data exchange "Report abnormal OBE"

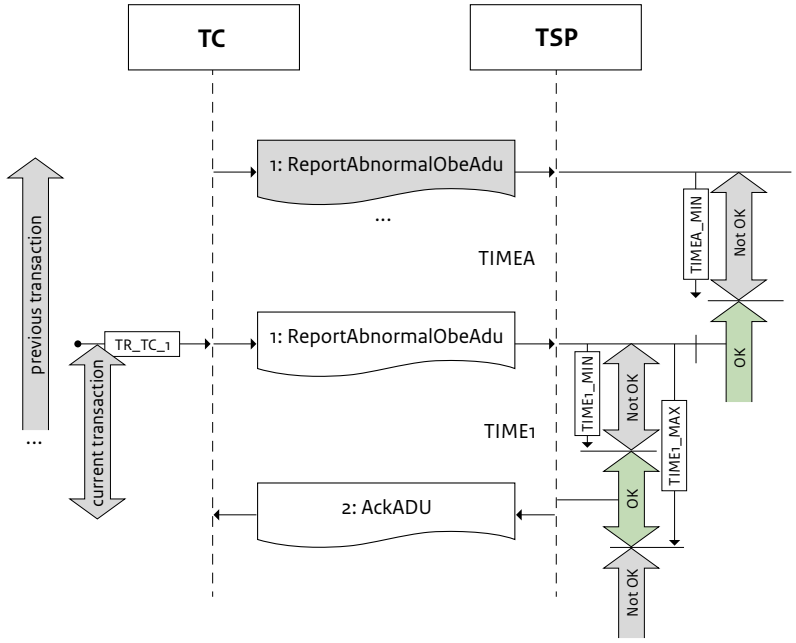
This exchange is used to report an abnormal detected OBE behaviour as a push message from the TC to the TSP.

A Report abnormal OBE message will have an user id containing at least a vehicle LPN or an obeId; and might also contain a Pan.

2.6.1 Specification "ReportAbnormalObeADU"

Seq	Qty	Lvl	Data item	Impl.	Value
1-20			See: 1.8 Message structure – InfoExchange data items		
21	1	2	reportAbnormalObeAdus		
22	1..n	3	ReportAbnormalObeAdu		
23	1	4	aduIdentifier	unsignedLong	
24	1	4	userId		
25	0	5	pan	hexBinary 10	
26	0	5	licencePlateNumber		
27	1	6	countryCode	Bitstring 10	
28	1	6	alphabetIndicator	token	Value: latinAlphabetNo1
29	1	6	licencePlateNumber	hexBinary	
30	0	5	obeId		
31	1	6	manufacturerId	unsignedShort	
32	1	6	equipmentObuId	hexBinary	
33	1	4	dateAndTime	DateTime	
34	1..n	4	abnormalObeReason		
35	1	5	abnormalObeReasonCode	unsignedByte	
36	1	5	additionalInformation	string	
37	1	4	actionCode	unsignedByte	Value: Send (0)

2.6.2 Sequence diagram



Sequence diagram of the REPORTABNORMALOBE transaction from EN 16986:2024.

2.6.3 Timing parameters

The table below contains the relevant timing parameters for REPORTABNORMALOBE for HGV Charge.

Timing	Value	More information
TIMEA_MIN	0	
TIME1_MIN	0	
TIME1_MAX	24 hours	

2.6.4 Specific requirements

There are some specific requirements for the exchange of an Report abnormal OBE message.

- An InfoExchange contains exactly one ReportAbnormalObeADU.

The transmission of a report abnormal OBE by TC is considered successful only after it has been acknowledged by TSP with an AckADU with an 'ApduOk' (where no single issue/error has occurred in any ADU).

2.7 Data exchange "Anonymous movement data" (obsolete)

This section is no longer in use. It is kept for numbering consistency only.

2.8 Data exchange “Efc Context Data”

This exchange is used to provide contextdata. The TC sends an Efc Context Data message to TSP. TSP uses context data for map matching as well as fee calculation.

2.8.1 Specification “EfcContextDataADU”

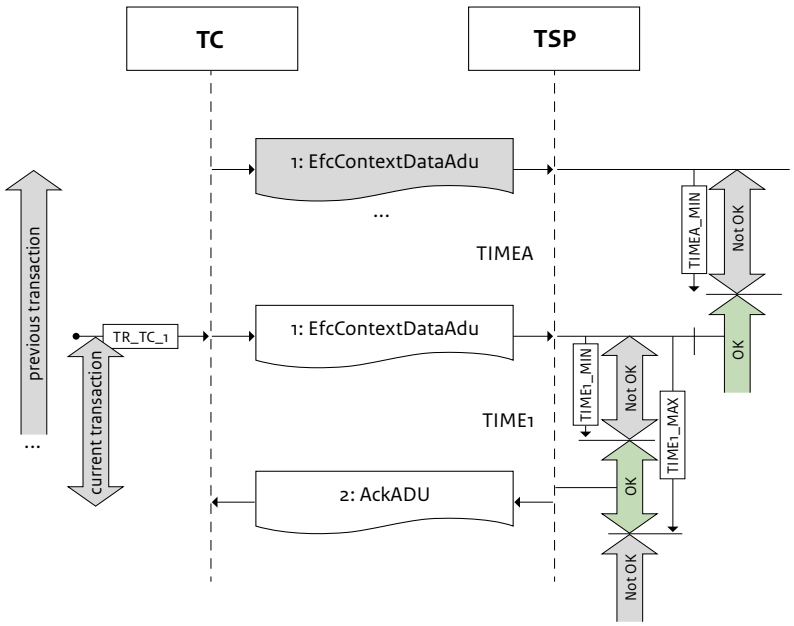
Seq	Qty	Lvl	Data item	Impl.	Value
1-20			See: 1.8 Message structure – InfoExchange data items		
21	1	2	efcContextDataAdus		
22	1..n	3	EfcContextDataAdu		
23	1	4	aduIdentifier	unsignedLong	
24	0	4	tollContextProperties		
25	0	5	tollContextOverview		
26	1	6	tollContext		
27	1	7	countryCode	Bitstring 10	
28	1	7	providerIdentifier	unsignedShort	
29	1	6	tollContextPartitions	string	
30	0	6	tollContextName	string	
31	1	6	tollContextOverviewVersion		
32	1	7	version	hexBinary	
33	1	7	validFrom	DateTime	
34	0	5	tariffTable		
35	1..n	6	TariffTable		
36	1	7	applicablePartitions	string	
37	1	7	tariffs		
38	1..n	8	Tariff		
39	1	9	tariffClass		
40	1	9	chargeUnit		
41	1	10	Distance		
42	1	11	distanceValue	unsignedInt	
43	1	11	distanceUnit	string	Value = kilometers (0)
44	1	9	roundingRuleForCharge-Unit	unsignedByte	Value: No rounding (0)
45	1	9	basicFeePerChargeUnit	unsignedInt	
46	1	9	roundingRuleForChargeUnit	unsignedByte	Value: No rounding (0)
47	0	9	infrastructureFee-PerChargeUnit	unsignedInt	

Seq	Qty	Lvl	Data item	Impl.	Value
48	0	9	externalCostsPerCharge-Unit		
49	0	10	airPollutionFeePerCharge-Unit	unsignedInt	
50	0	10	co2FeePerChargeUnit	unsignedInt	
51	1	7	defaultCurrency	hexBinary 2	Value: 1/10 Euro-cent (3978)
52	0	7	typeOfTariff	unsignedByte	Value: fee (1)
53	1	7	tariffTableVersion		
54	1	8	version	hexBinary	
55	1	8	validFrom	DateTime	
56	0	5	tariffClassDefinition		
57	1	6	tariffClasses		
58	1..n	7	TariffClass		
59	1	8	tariffClassId	unsignedInt	
60	0	8	tariffClassDescription	string	
61	1	8	localVehicleClasses	unsignedShort	
62	0	8	locationClasses	unsignedInt	
63	1	6	tariffClassDefinitionVersion		
64	1	7	version	hexBinary 10	
65	1	7	validFrom	DateTime	
66	0	5	localVehicleClassDefinition		
67	1	6	localVehicleClasses		
68	1..n	7	localVehicleClass		
69	1	8	localVehicleClassId	unsignedShort	
70	0	8	localVehicleClassDescription	string	
71	1	8	nominalVehicleParameters		
72	1	9	vehicleClasses	unsignedInt	Value = 0 (Not in use)
73	0	9	euroValues	string	
74	1	9	co2Classes	string	
75	0	8	ordinalVehicleParameters		
76	0	9	vehicleTrainMaximum-Weight		
77	0..n	10	VehicleTrainMaximum-WeightRange		
78	0	11	lowerLimit	unsignedShort	
79	0	11	upperLImit	unsignedShort	
80	0	6	localVehicleClassVersion		

Seq	Qty	Lvl	Data item	Impl.	Value
81	1	7	version	hexBinary	
82	1	7	validFrom	DateTime	
83	0	4	tollContextType		
84	1	5	generalContext		
85	1..n	6	tollContextPartitionOverviews		
86	1	7	TollContextPartitionOverview		
87	1	8	tollContextPartitionId	unsignedByte	
88	0	8	tollContextPartitionName	string	
89	1	8	tollContextPartitionType	unsignedByte	Value = roadSectionPricing (0)
90	1	8	operationalStatus		
91	1	9	startsOperationAt	DateTime	
92	0	9	stopsOperationAt	DateTime	
93	1	8	timeZone	short	
94	0	8	dstInformation		
95	0	9	dstOffset	short	
96	1	9	dstStartDate	DateTime	
97	1	9	dstEndDate	DateTime	
98	1	8	tollContextPartition-Overview Version		
99	1	9	version	hexBinary	
100	1	9	validFrom	DateTime	
101	0	6	tollContextPartitionLayouts		
102	1..n	7	TollContextPartitionLayout		
103	1	8	tollContextPartitionId	unsignedByte	
104	1	8	layoutDescription		
105	1	9	sectionLayoutDescription		
106	1	10	sections		
107	1..n	11	Section		
108	1	12	chargeObjectId		
109	0	13	chargeObjectOperator		
110	1	14	countryCode	Bitstring 10	
111	1	14	providerIdentifier	unsignedShort	
112	1	13	chargeObject-Designation	unsignedInt	
113	0	12	chargeObjectName	string	
114	0	12	tollPath		

Seq	Qty	Lvl	Data item	Impl.	Value
115	1	13	Link		
116	1	14	linkId	unsignedInt	
117	1	14	startPoint		
118	1	15	absolutePoint Coordinates		
119	1	16	longitude	integer	
120	1	16	latitude	integer	
121	0	14	intermediatePoints		
122	1..n	15	Point		
123	1	16	absolutePoint Coordinates		
124	1	17	longitude	integer	
125	1	17	latitude	integer	
126	1	14	endPoint		
127	1	15	absolutePoint Coordinates		
128	1	16	longitude	integer	
129	1	16	latitude	integer	
130	0	12	liabilityRules		
131	0	13	minimumUsage	unsignedShort	*1 = 0.1%, 1000 = 100%*
132	1	12	chargeDistance		
133	1	13	distanceValue	unsignedInt	
134	1	13	distanceUnit	unsignedByte	Value = ‘meters’ or ‘2’
135	1	12	locationClass		
136	1	8	tollContextPartitionVersion		
137	1	9	version	hexBinary	
138	1	9	validFrom	DateTime	
139	0	6	tollContextVersion		
140	1	7	version	hexBinary	
141	1	7	validFrom	DateTime	
142	0	4	actionCode	unsignedByte	Value: Send (0), Revoke (1)

2.8.2 Sequence diagram



Sequence diagram of the EFCCONTEXTDATA transaction from EN 16986:2024.

2.8.3 Timing parameters

The table below contains the relevant timing parameters for EFCCONTEXTDATA for HGV Charge.

Timing	Value	More information
TR_TC_1		A TC may initiate a EFCCONTEXTDATA transaction at any time after TIMEA_MIN
TIMEA_MIN	1 minute	
TIME1_MIN	0	
TIME1_MAX	24 hours	

2.8.4 Specific requirements

There are some specific requirements for the exchange of an Efc context data message.

- An InfoExchange contains exactly one EfcContextDataADU.
- The transmission of an Efc context data message by TC is considered successful only after it has been acknowledged by TSP with an AckADU with an 'apduOk' (where no single issue/error has occurred in any ADU).

2.9 Data exchange “Trust object”

This exchange is used to exchange keys of data exchange security. A trust object message can be sent by either TC or TSP to respectively TSP or TC.

2.9.1 Specification “TrustObjectADU”

Seq	Qty	Lvl	Data item	Impl.	Value
1-20			See: 1.8 Message structure – InfoExchange data items		
21	1	2	trustObjectAdus		
22	1..n	3	TrustObjectAdu		
23	1	4	aduIdentifier	unsignedLong	
24	0	4	replacedTrustObject	unsignedLong	
25	1..n	4	purposesOfTrustObject		
26	1	5	TrustObjectPurpose	unsignedByte	
27	0	4	startValidity	DateTime	
28	0	4	endValidity	DateTime	
29	1	4	trustObjectStatus	unsignedByte	
30	0	4	trustObject		
31	1	5	certificate		
32	1	7	certificateType	unsignedByte	
33	1	7	certificate	token	
34	0	4	actionCode	unsignedByte	

2.9.2 Response

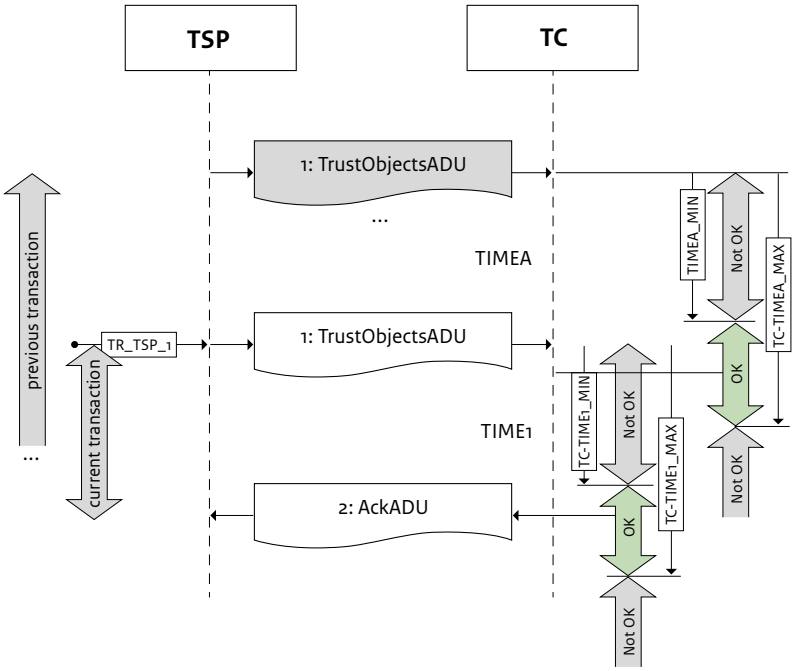
An AckADU is sent by TC in response to an InfoExchange message containing a TrustObjectADU that is sent by TSP. See: section 2.1.1 AckADU.

2.9.3 Validation and errors

An trustObjectADU is functionally validated. If these validations are not successful, an AckADU with an ‘ApduOk’ wil be returned containing one or more of the following errors.

#	Check	Validation	Error (if not successful)
1 CERTAdu.F1	certificate-Type	Value of trustObjectAdu.trustObject. certificate.certificateType must be: certTspHTTPS (5) or certTspNSIG (7)	3000 - unexpected certificate type
2 CERTAdu.F2	certificate	trustObjectAdu.trustObject.certificate. certificate must be a valid certificate for this certificate type	trustObjectUnreadable (101) : invalid certificate
3 CERTAdu.F3	Expirydate	Expiry date of the certificate must be in the future *Note: Expiry date is part of the certificate itself*	trustObjectExpired (100) : certificate has expired

2.9.4 Sequence diagram – TSP to TC

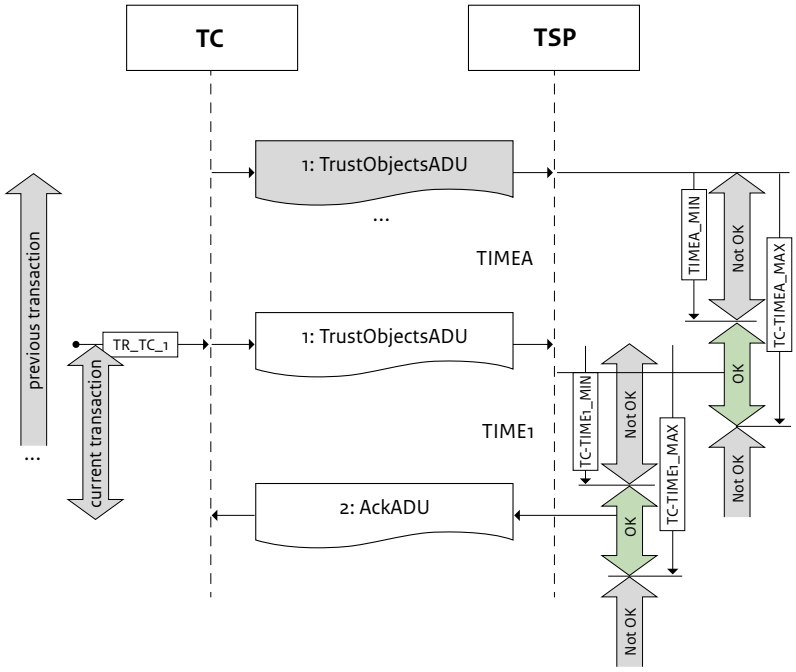


2.9.5 Timing parameters – TSP to TC

The table below contains the relevant timing parameters for TRUSTOBJECTS for HGV Charge.

Timing	Value	More information
TSP-TIMEA_MIN	0	
TSP-TIME1_MIN	0	
TSP-TIME1_MAX	24 hours	

2.9.6 Sequence diagram – TC to TSP



2.9.7 Timing parameters – TC to TSP

The table below contains the relevant timing parameters for TRUSTOBJECTS for HGV Charge.

Timing	Value	More information
TC-TIMEA_MIN	0	
TC-TIME1_MIN	0	
TC-TIME1_MAX	24 hours	

2.9.8 Specific requirements

There are some specific requirements for the exchange of an Trust object message.

- An InfoExchange contains exactly one TrustObjectADU.
- The transmission of a Trust objects message by either TSP or TC is considered successful only after it has been acknowledged by respectively TC or TSP with an AckADU with an 'apduOk' (where no single issue/error has occurred in any ADU).
- Certificates must be issued by a trusted Class III Certificate Authority or equivalent.

The certificate token should be specified as a UTF-8 byte-encoded PEM string containing a certificate bundle. The bundle contains the (leaf) certificate, as well as any intermediate certificates and the root. Each certificate is encoded according to RFC 7468, section 5.1. The bundle is created by concatenating PEM-encoded certificates together as suggested in RFC 7468, section 1, such that the leaf certificate is the topmost certificate and the CA root is at the bottom.

The string representation of the resulting bundle has the following structure:

-----BEGIN CERTIFICATE-----
<DER-encoded leaf certificate>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<DER-encoded intermediate certificate 1>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<DER-encoded intermediate certificate 2>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<DER-encoded root certificate>
----END CERTIFICATE-----

The TC will attempt to process other representations of the certificate, but no guarantee is made that processing will succeed.

2.10 Data exchange "Request for Toll Declaration"

The TC requests the TSP to send a Toll Declaration message to the TC. A request for a Toll Declaration can either be made for a specific toll trip or period of time.

2.10.1 Specification "RequestADU"

Seq	Qty	Lvl	Data item	Impl.	Value collection
1-20			See: 1.8 Message structure – InfoExchange data items		
21	1	2	requestAdus		
22	1..n	3	RequestAdu		
23	1	4	tollDeclarationRequest		
24	1	5	requestedAduType	unsignedByte	Value: tollDeclarationAdu (7)
25	0	5	aduIdentifier	unsignedLong	
26	0	5	userId		
27	0	6	pan	hexBinary 10	
28	0	6	licencePlateNumber		
29	1	7	countryCode	Bitstring 10	
30	1	7	alphabetIndicator	token	Value: latinAlphabetNo1
31	1	7	licencePlateNumber	hexBinary	
32	0	7	obeID		
33	1	8	manufacturerId	unsignedShort	
34	1	8	equipmentObuId	hexBinary	
35	0	5	startTime	DateTime	
36	0	5	endTime	DateTime	
37	1	5	actionCode	unsignedByte	Value: Send (0)

2.10.2 Specification “TollDeclarationADU”

Seq	Qty	Lvl	Data item	Impl.	Value collection
1-20			See: 1.8 Message structure – InfoExchange data items		
21	1	2	tollDeclarationAdus		
22	1..n	3	TollDeclarationAdu		
23	1	4	aduIdentifier	unsignedLong	
24	1	4	tollDeclarationId		
25	1	5	issuerId		
26	1	6	countryCode	Bitstring 10	
27	1	6	providerIdentifier	unsignedShort	
28	1	5	declarationId	unsignedLong	
29	1	4	chargeReport		
30	0..n	5	ChargeReport		
31	1	6	obeId		
32	1	7	manufacturerId	unsignedShort	
33	1	8	equipmentObuId	hexBinary	
34	1	6	vehicleLPNr		
35	1	7	countryCode	Bitstring 10	
36	1	7	alphabetIndicator	token	Value: latinAlphabetNo1
37	1	7	licencePlateNumber	hexBinary	
38	1	6	paymentMeans		
39	1	7	personalAccountNumber		
40	1	8	paymentMeansExpiryDate		
41	1	8	year		
42	1	8	month		
43	1	7	day		
44	1	6	paymentUsageControl		
45	1	6	serviceProviderContract		
46	1	7	contractProvider		
47	1	8	countryCode	Bitstring 10	
48	1	8	providerIdentifier	unsignedShort	
49	1	7	typeOfContract	hexBinary 2	
50	1	7	contextVersion	unsignedInt	
51	1	6	tollContext		
52	1	7	countryCode	Bitstring 10	
53	1	7	providerIdentifier	unsignedShort	
54	1	6	reportPeriod		
55	1	7	beginOfPeriod	DateTime	
56	1	7	endOfPeriod	DateTime	
57	1	6	appliedTollContextVersion	hexBinary	
58	1	6	usageStatementList		

Seq	Qty	Lvl	Data item	Impl.	Value collection
59	1	7	usageStatement		
60	0	8	listOfChargeObjects		
61	1..n	9	DetectedChargedObject		
62	1	10	chargeObjectId		
63	1	11	chargeObjectOperator		
64	1	12	countryCode	Bitstring 10	
65	1	12	providerIdentifier	unsignedShort	
66	1	11	chargeObjectDesignation *The unique identifier of a road section (charge-object-designation from EfcContextData message)*		
67	1	10	timeWhenUsed	DateTime	
68	1	6	currentTariffClass		
69	1	7	tariffClassId	unsignedInt	
70	0	7	localVehicleClassId	unsignedShort	
71	1	6	chargeReportCounter	unsignedShort	
72	1	4	actionCode	unsignedByte	Value: Send (0)
73	1	4	geoJsonDataObject* *CDATA method must be used for geojson file content*	string *Unlimited*	

*) *For more details about the geojson-file. See: 2.10.7 Specific requirements.*

2.10.3 Response

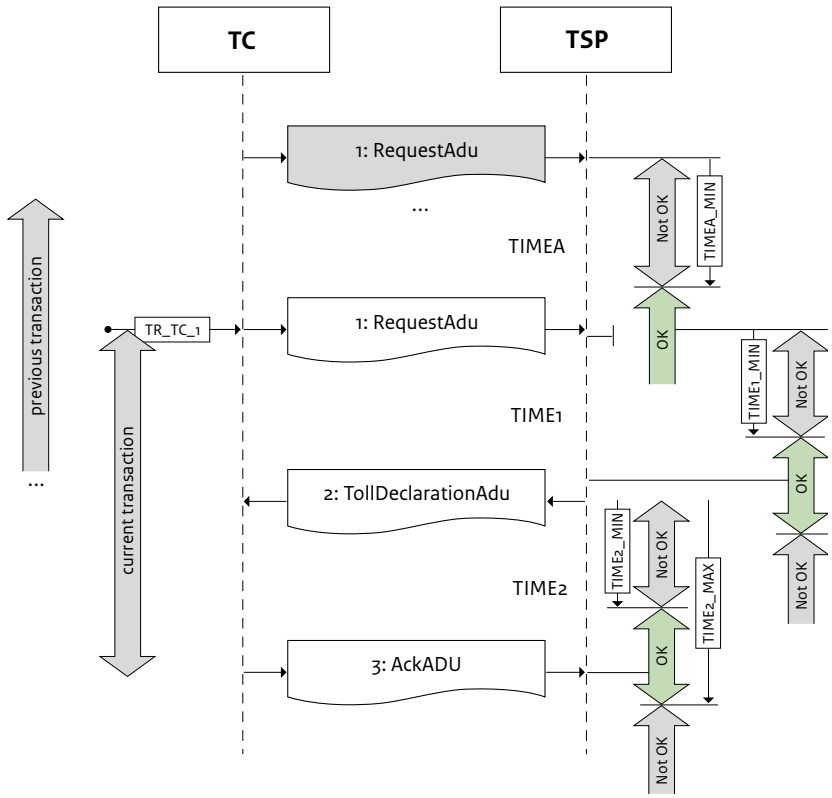
An AckADU is sent by TC in response to an InfoExchange message, containing one or more TollDeclarationADU's, that is sent by TSP. See: section 2.1.1 AckADU.

2.10.4 Validation and errors

A TollDeclarationADU is functionally validated. The following errors will be returned, if validations are not successful.

#	Check	Validation	Error (if not successful)
1 TDCLApci.F1	APCI.inResponse-ToApduId	Must be present	3000 - inResponseToApduId missing
2 TDCLRec.F1	issuerId	Must be equal to apduOriginator from apci	3000 - issuerId mismatch
3 TDCLRec.F2	countryCode	Value of chargereport.vehicleLP-NR.countryCode must be a valid country (ISO 3166-1, Alpha-2)	3000 - Invalid licence plate countryCode
4 TDCLRec.F3	licencePlateNumber	Value of alphabetIndicator must be latinAlphabetNo1	3000 - exceptionListsUserId-Rejected: invalid alphabetIndicator in licencePlateNumber
5 TDCLRec.F4	geoJsonDataObject	Geojson file structure must be in conformity with 2.10.7.1 Specification "geojson" (as part of a Toll declaration)	3000 – Invalid geojson

2.10.5 Sequence diagram



Sequence diagram of the TOLLDECLARATION transaction from EN 16986:2024.

2.10.6 Timing parameters

The table below contains the relevant timing parameters for TOLLDECLARATION for HGV Charge.

Timing	Value	More information
TIMEA_MIN	0	Multiple toll declaration interactions can be started in parallel
TIME1_MIN	0	
TIME1_MAX[*]	1 minute	First three months after a toll trip has been reported
TIME2_MIN	0	
TIME2_MAX	24 hours	

[*] *Timing parameter for TIME1_MAX*
A response to Request for Toll declaration must be received within one minute for each toll trip. This time limit must be upheld at least three months after the related toll trip or toll trips have been reported through a Billing details message. E.g. if a toll trip is reported on 19 March 2025, the subsequent Toll declaration must be delivered within one minute until at least 19 June 2025. Beyond these three months, no specific time limit is set.

2.10.7 Specific requirements

There are some specific requirements for the exchange of a Toll Declaration message.

- An interaction starts with a request being sent by TC to TSP for a TollDeclaration. This is a top-up for the SectionAutonomous profile.
- A Toll Declaration message must contain a geojson file as an attachment in the geoJsonDataObject[2] data element. The CDATA method must be used for the content of the geoJsonDataObject data element.

[2] *The geoJsonDataObject data element is a Top-up that is placed on equal level with the TollDeclarationADU or ADU's. This is done in order to prevent multiple sending of the same geojson file in one Toll declaration message when there are more then one TollDeclarationADU's.*

- The transmission of a Toll declaration message by TC is considered successful after it has been acknowledged by TSP with an AckADU with an 'apduOk' (no single issue/error has occured).

Geojson-file

GeoJSON is an open standard format for encoding a variety of geographic data structures, which is designed for representing simple geographical features. A geojson file can easily be viewed in a browser application, thus making it very usefull to have a quick and clear insight into the movements of a service user (vehicle).

A visual representation of a geojson-file

In the table below a specification of a geojson-file is described. An example of a geojson-file can be found in Appendix A – Example of a geojson file.

Seq	Qty	Lvl	Data item	Description	Impl.	Value
1	1	1	type			Value: Feature-Collection
2	1	1	features			
3	1	2	type			Value: Feature
4	1	2	geometry	Geographical part of the road network consisting of several positions; points which form a linestring		
5	1..n	3	coordinates	A position measured in longitude and latitude		

Seq	Qty	Lvl	Data item	Description	Impl.	Value
6	1	3	type			Value: Linestring
7	1	2	properties			
8	1	3	HgvCharge	Indicator if a geographical part of the road network is map matched with a chargeable road section of the HGV Charge EfcContextData.	Boolean	
9	0..1	3	id	If HgvCharge is true, id must be present. This is the identifier of a chargeable road section or road section (link id) from the EfcContectData message. It must also match	unsigned-Long	
10	0..1	3	length	If HgvCharge is true, length must be present. This value must match the distanceValue of the charge object from the EfcCon-textData message.	unsigned-Long	
11	0..1	3	name	If HgvCharge is true, name must be present. This value must match the charge-ObjectName of the charge object from the EfcContextData message.	String 200	
12	1	3	startDateTime	Moment when the vehicle entered this part.	DateTime	
13	1	3	endDateTime	Moment when the vehicle left this part.	DateTime	

2.11 Data exchange “Request for User Details”

The TC requests the TSP to send an User Details message to the TC. A request for a User details message will be made for a specific user.

2.11.1 Specification “RequestADU”

Seq	Qty	Lvl	Data item	Impl.	Value collection
1-20			See: 1.8 Message structure – InfoExchange data items		
21	1	2	requestAdus		
22	1..n	3	RequestAdu		
23	1..n	4	userDetailsRequest		
24	1	5	requestedAduType	unsignedByte	Value: ProvideUserDetails Adu (12)
25	0	5	aduIdentifier	unsignedLong	
26	1	5	userId		
27	0	6	pan	hexBinary 10	
28	0	6	licencePlateNumber		
29	1	7	countryCode	Bitstring 10	
30	1	7	alphabetIndicator	token	

Seq	Qty	Lvl	Data item	Impl.	Value collection
31	1	7	licencePlateNumber	hexBinary	
32	0	7	obeID		
33	1	8	manufacturerId	unsignedShort	
34	1	8	equipmentObuId	hexBinary	
35	1	5	listOfParametersRequested		For values, see: below
36	1	5	userDetailsRequestReason	unsignedByte	
37	0	5	period		
38	1	6	beginOfPeriod	DateTime	
39	1	6	endOfPeriod	DateTime	
40	1	5	actionCode	unsignedByte	Value: Send (0)

Values for data item listOfParametersRequested

Values which are in use for the Dutch HGV Charge implementation:

- contractValidity (2),
- efcContextMark (4),
- environmentalCharacteristics (5),
- engineCharacteristics (6),
- equipmentObuId (7),
- equipmentStatus (8),
- personalAccountNumber (12),
- validityOfContract (15),
- vehicleClass (17),
- vehicleDimensions (18),
- vehicleLicencePlateNumber (19),
- userPostalAddress (26),
- preferredUserLanguage (27),
- cO2EmissionValueLoad (29),
- initialVehicleRegistrationDate (30),
- euVehicleGroup (31),
- engineDetails (32),
- cabType (33),
- euCO2EmissionClass (34)

2.11.2 Specification “ProvideUserDetailsADU”

Seq	Qty	Lvl	Data item	Impl.	Value collection
1-20			See: 1.8 Message structure – InfoExchange data items		
21	1	2	provideUserDetailsAdus		
22	1..n	3	ProvideUserDetailsAdu		
23	1	4	aduIdentifier	unsignedLong	
24	1	4	originalUserIdRequest		
25	0	5	pan	Bitstring 10	
26	0	5	contractSerialNumber	unsignedInt	
27	0	5	licencePlateNumber		
28	1	6	countryCode	Bitstring 10	
29	1	6	alphabetIndicator	token	Value: latinAlphabet-No1
30	1	6	licencePlateNumber	hexBinary	
31	0	5	obeID		
32	1	6	manufacturerId	unsignedShort	
33	1	6	equipmentObuId	hexBinary	
34	1	4	userId		
35	0	5	pan	Bitstring 10	
36	0	5	contractSerialNumber	unsignedInt	
37	0	5	licencePlateNumber		
38	1	6	countryCode	Bitstring 10	
39	1	6	alphabetIndicator	token	Value: latinAlphabet-No1
40	1	6	licencePlateNumber	hexBinary	
41	0	5	obeID		
42	1	6	manufacturerId	unsignedShort	
43	1	6	equipmentObuId	hexBinary	
44	1	4	statusFlag	unsignedByte	
45	0	4	listOfUserParameters		
46	0..n	5	userParameterResponse		
47	0	6	requestedUser Parameter	unsignedByte	
48	1	6	userParameterResponse		
49	0	7	contractSerialNumber	unsignedInt	
50	0	7	contractValidity		
51	1	8	contractRestrictions	hexBinary 4	
52	1	8	contractExpiryDate		
53	1	9	year	unsignedShort	
54	1	9	month	unsignedByte	
55	1	9	day	unsignedByte	
56	0	7	driverCharacteristics		
57	1	8	driverClass	unsignedByte	
58	1	8	tripPurpose	unsignedByte	
59	0	7	efcContextMark		

Seq	Qty	Lvl	Data item	Impl.	Value collection
60	1	8	contractProvider		
61	1	9	countryCode	Bitstring 10	
62	1	9	providerIdentifier	unsignedShort	
63	1	8	typeOfContract	hexBinary 2	
64	1	8	contextVersion	unsignedByte	
65	0	7	environmentalCharacteristics		
66	1	8	euroValue	unsignedByte	
67	1	8	copValue	unsignedByte	
68	0	7	engineCharacteristics	unsignedByte	
69	0	7	equipmentObuId	hexBinary	
70	0	7	equipmentStatus	Bitstring 16	
71	0	7	paymentMeans		
72	1	8	personalAccountNumber	Hexbinary 10	
73	1	8	paymentMeans ExpiryDate		
74	1	9	year	unsignedShort	
75	1	9	month	unsignedByte	
76	1	9	day	unsignedByte	
77	1	8	paymentMeansUsageControl	hexBinary 2	
78	0	7	paymentMeansBalance		
79	0	8	positive	unsignedInt	
80	0	8	negative	unsignedInt	
81	0	7	payUnit	hexBinary 2	
82	0	7	personalAccountNumber	Hexbinary 10	
83	0	7	provider		
84	1	8	countryCode	Bitstring 10	
85	1	8	providerIdentifier	unsignedShort	
86	0	7	receiptContract		
87	1	8	sessionContractProvider		
88	1	9	countryCode	Bitstring 10	
89	1	9	providerIdentifier	unsignedShort	
90	1	8	sessionTypeOfContract	hexBinary 2	
91	1	8	sessionContractSerialNumber	unsignedInt	
92	0	7	validityOfContract		
93	1	8	issuerRestrictions	hexBinary 2	
94	1	8	contractExpiryDate		
95	1	9	year	unsignedShort	
96	1	9	month	unsignedByte	
97	1	9	day	unsignedByte	
98	0	7	vehicleAuthenticator	hexBinary	
99	0	7	vehicleClass	unsignedByte	
100	0	7	vehicleDimensions		
101	1	8	vehicleLengthOverall	unsignedByte	
102	1	8	vehicleHeigthOverall	unsignedByte	
103	1	8	vehicleWidthOverall	unsignedByte	

Seq	Qty	Lvl	Data item	Impl.	Value collection
104	0	7	vehicleLicencePlateNumber		
105	1	8	countryCode	Bitstring 10	
106	1	8	alphabetIndicator	token	Value: latinAlphabet-No1
107	1	8	licencePlateNumber	hexBinary	
108	0	7	vehicleIdentificationNumber		
109	1	8	vin	string	
110	1	8	fill	Bitstring 9	
111	0	7	vehicleWeightLaden	unsignedShort	
112	0	7	vehicleWeightLimits		
113	1	8	vehicleMaxLadenWeight	unsignedShort	
114	1	8	vehicleTrainMaximumWeight	unsignedShort	
115	1	8	vehicleWeightUnladen	unsignedShort	
116	0	7	vehicleAxles		
117	1	8	vehicleFirstAxleHeight	unsignedByte	
118	1	8	vehicleAxlesNumber		
119	1	9	tyreConfiguration	unsignedByte	
120	1	9	numberOfAxles		
121	1	10	trailerAxles	unsignedByte	
122	1	10	tractorAxles	unsignedByte	
123	0	7	exhaustEmissionValues		
124	1	8	unitType	unsignedByte	
125	1	8	emissionCo	unsignedShort	
126	1	8	emissionHc	unsignedShort	
127	1	8	emissionNox	unsignedShort	
128	1	8	emissionHcNox	unsignedShort	
129	0	7	dieselEmissionValues		
130	1	8	particulate		
131	1	9	unitType	unsignedByte	
132	1	9	value	unsignedShort	
133	1	8	absorptionCoeff	unsignedShort	
134	0	7	userPostalAddress		
135	0	8	addresseeRoleDescriptor	string 10	
136	0	8	organisationName	string 256	
137	0	8	organisationUnit	string 256	
138	0	8	function	string 256	
139	0	8	formOfAddress	string 50	
140	0	8	qualification	string 50	
141	0	8	surname	string 256	
142	0	8	givenName	string 256	
143	0	8	deliveryServicePoint	string 256	
144	0	8	thoroughfare	string 256	
145	0	8	postCode	string 10	

Seq	Qty	Lvl	Data item	Impl.	Value collection
146	0	8	town	string 256	
147	0	8	region	string 256	
148	0	8	country	string 256	
149	0	8	phoneInternationalDiallingCode	string 5	
150	0	8	phoneDiallingCode	string 10	
151	0	8	phoneSubscriberNumber	string 20	
152	0	8	mobileInternationalDialling-Code	string 5	
153	0	8	mobileDiallingCode	string 10	
154	0	8	mobileSubscriberNumber	string 20	
155	0	8	faxInternationalDiallingCode	string 5	
156	0	8	faxDiallingCode	string 10	
157	0	8	faxSubscriberNumber	string 20	
158	0	8	email	string 256	
159	0	8	companyRegistrationNumber	string 30	
160	0	8	taxIdentifier	string 50	
161	0	8	vatIdentifier	string 30	
162	0	7	preferredUserLanguage	string	
163	0	7	equipmentIccId	hexBinary	
164	0	7	co2EmissionValueLoad	unsignedInt	
165	0	7	initialVehicleRegistrationDate		
166	1	8	year	unsignedShort	
167	1	8	month	unsignedByte	
168	1	8	day	unsignedByte	
169	0	7	euVehicleGroup		
170	1	8	mainEuVehicleGroup	unsignedByte	
171	0	8	subGroup	string 2	
172	0	7	engineDetails		
173	1	8	engineCapacity	unsignedShort	
174	1	8	enginePower	unsignedShort	
175	0	7	cabType	unsignedByte	
176	0	7	euCO2EmissionClass	unsignedByte	
177	0	6	userParameterStatus	unsignedByte	
178	1	4	actionCode	unsignedByte	

2.11.3 Response

An AckADU is sent by TC in response to an InfoExchange message, containing one or more ProvideUserDetailsADU's, that is sent by TSP. See: section 2.1.1 AckADU.

2.11.4 Validation and errors

A ProvideUserDetailsADU is functionally validated. The following errors will be returned, if validations are not successful.

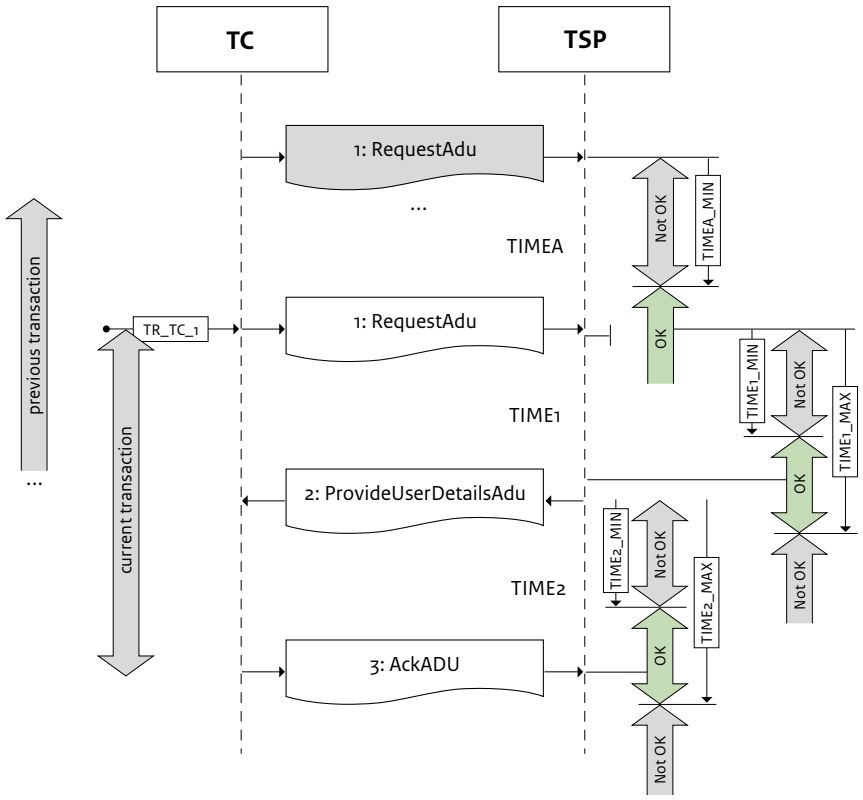
#	Check	Validation	Error (if not successful)
1 PUDTApci.F1	APCI. inResponse-ToApduId	Must be present	3000 - inResponseToApduId missing
2 PUDTRec.F1	requested-User-Parameter	The data item for the requestedUserPa-rameter must be present in the UserParameterResponseDetails*	
3 PUDTRec.F2	countryCode	Value of chargereport.vehicleLPNR. countryCode must be a valid country (ISO 3166-1, Alpha-2)	3000 - Invalid licence plate countryCode
4 PUDTRec.F3	licencePlate-Number	Value of alphabetIndicator must be latinAlphabetNo1	3000 - UserDetailsUserIdRejected: invalid alphabetIndicator in licencePlateNumber

*) *The following requestedUserParameters are in use:*

- *contractValidity* (2),
- *efcContextMark* (4),
- *environmentalCharacteristics* (5),
- *engineCharacteristics* (6),
- *equipmentObuId* (7),
- *equipmentStatus* (8),
- *personalAccountNumber* (12),
- *validityOfContract* (15),
- *vehicleClass* (17),
- *vehicleDimensions* (18),
- *vehicleLicencePlateNumber* (19),
- *userPostalAddress* (26),
- *preferredUserLanguage* (27),
- *cO2EmissionValueLoad* (29),
- *initialVehicleRegistrationDate* (30),
- *euVehicleGroup* (31),
- *engineDetails* (32),
- *cabType* (33),
- *euCO2EmissionClass* (34)

Other requestedUserParameters will not be used.

2.11.5 Sequence diagram



Sequence diagram of the USERDETAILS transaction from EN 16986:2024.

2.11.6 Timing parameters

The table below contains the relevant timing parameters for USERDETAILS for HGV Charge.

Timing	Value	More information
TIMEA_MIN	0	Multiple user details interactions can be started in parallel
TIME1_MIN	0	
TIME1_MAX	1 minute	Up to three months after a whitelist entry has been ended
TIME2_MIN	0	
TIME2_MAX	24 hours	

2.11.7 Specific requirements

There are some specific requirements for the exchange of a Provide user details message.

- The transmission of a Provide user details message by TC is considered successful after it has been acknowledged by TSP with an AckADU with an ‘apduOk’ (no single issue/error has occured).

List of definitions and abbreviations

Abbreviation	Explanation
ADU	Application Data Unit; a data structure of a specific type
APCI	Application Protocol Control Information
APDU	Application Protocol Data Unit
Chargeable road section	A road section for which a Heavy Goods Vehicle Charge is owed when it is used or driven.
Chargeable network	The network consists of chains of connected chargeable road sections. The chargeable network is entered when leaving a non-chargeable road section and entering a chargeable road section. The chargeable network is exited when driving on a chargeable road section is followed by driving on a non-chargeable road section.
EETS	European Electronic Toll Service; an EETS provider is a Toll Service Provider.
EFC	Electronic Fee Collection
HGV Charge	Heavy Goods Vehicle Charge
LPN	License plate number
MSP	Main Service Provider; the MSP is a Toll Service Provider
OBE	On Board Equipment
Pan	Personal account number
RDW	The Dutch Vehicle Authority; the Toll Charger for the HGV Charge
TC	Toll Charger
Toll trip	A toll trip is a contiguous series of road sections starting with the first road section when a vehicle enters the chargeable road network and ends with: • the road section where the vehicle leaves the chargeable road network; • the road section where a new version of Efc Context Data comes into effect; • the road section where the status of the OBE is changed to a NoGo status (ISO 12813:213); or • the road section that has been registered 72 hours after the start of the toll trip.
TSP	Toll Service Provider
SU	Service user. User of a toll service that has concluded a service agreement for its vehicle with a toll service provider. A service user (and its vehicle) is identified by the unique combination of pan, LPN and Obe id.
Vrachtwagenheffing (VWH)	Heavy Goods Vehicle Charge (HGV Charge)
Zero-km notification	A Billing details message containing 0 (zero) chargeable kilometers, indicating a vehicle has driven on the Dutch road network without entering the chargeable network.

Annex A - Example of a geojson file

This appendix contains an example of geojson file content. This content should be the result of the map matching process. It contains movement data of the vehicle of a service user.

Visual representation of the geojson-file

It contains eight features:

- Six features represent chargeable road sections that have been identified as a result of the map matching process.
- The two other features represent the movements of the vehicle before and after entering the chargeable network.

The chargeable road sections are uniquely identified:

- id: 600138412 (name: Lelystad-Noord 11)
- id: 600138942 (name: Lelystad-Noord 11)
- id: 600138410 (name: Rijksweg A6)
- id: 600137671 (name: Rijksweg A6)
- id: 325398053 (name: Lelystad 10)
- id: 325398052 (name: Lelystad 10)

CDATA method

The data in geojson file obviously is not formatted as an XML, but as a JSON. This data must be inserted into the geoJsonDataObject of a TollDeclaration message. In order for the data to be read correctly, the CDATA method must be used for the content of the geoJsonDataObject data element

The visual representation in this appendix is generated from the data in the example below.

{“features”: [{“geometry”: {“coordinates”: [[5.50139135882043, 52.481239757167], [5.50196623310109, 52.4815341673705], [5.50238432348726, 52.4818007262613], [5.50244311744788, 52.4819757789822], [5.50233859485135, 52.4821428741114], [5.50180944920666, 52.4824372782718], [5.50186171050453, 52.482481040883 7], [5.50228633355346, 52.4827237236713], [5.5025541727064, 52.4825645875686]], “type”: “LineString”}, “properties”: {“endDateTime”: “2023-10-01T18:28:42Z”, “HgvCharge”: false, “startDateTime”: “2023-10-01T18:25:43Z”}, “type”: “Feature”}, {“geometry”: {“coordinates”: [[5.530814, 52.54707], [5.530855, 52.546892], [5.530889, 52.546734], [5.53105, 52.545804], [5.531156, 52.545656]], “type”: “LineString”}, “properties”: {“endDateTime”: “2023-10-01T18:29:03Z”, “HgvCharge”: true, “id”: 600138412, “length”: 159.805705968478, “name”: “Lelystad-Noord 11”, “startDateTime”: “2023-10-01T18:28:42Z”}, “type”: “Feature”}, {“geometry”: {“coordinates”: [[5.527952, 52.548509], [5.528027, 52.54877], [5.528239, 52.549335], [5.528389, 52.549523], [5.528506, 52.549621], [5.52875, 52.549735], [5.529102, 52.549825], [5.529463, 52.549835], [5.529815, 52.549775], [5.530041, 52.549689], [5.530337, 52.549517], [5.530489, 52.549259], [5.530553, 52.548886], [5.530687, 52.54779], [5.530814, 52.54707]], “type”: “LineString”}, “properties”: {“endDateTime”: “2023-10-01T18:29:57Z”, “HgvCharge”: true, “id”: 600138942, “length”: 547.608550341747, “name”: “Lelystad-Noord 11”, “startDateTime”: “2023-10-01T18:29:03Z”}, “type”: “Feature”}, {“geometry”: {“coordinates”: [[5.531156, 52.545656], [5.531308, 52.544401], [5.531475, 52.542958], [5.531623, 52.541623], [5.531773, 52.540197], [5.531904, 52.538877], [5.532034, 52.537485], [5.5322, 52.535607], [5.532281, 52.534627], [5.532366, 52.533542], [5.532447, 52.532441], [5.53247, 52.532082], [5.532549, 52.53068], [5.53256, 52.5305], [5.532619, 52.529782], [5.53264, 52.529504], [5.532696, 52.528606], [5.532782, 52.527077], [5.532837, 52.526006], [5.532864, 52.525467], [5.532938, 52.523772], [5.532998, 52.522216], [5.533034, 52.521138], [5.533081, 52.519473], [5.533119, 52.517791], [5.533138, 52.516715], [5.533162, 52.514962], [5.533173, 52.512945], [5.533172, 52.511965], [5.533161, 52.511426], [5.533147, 52.511067], [5.533127, 52.510707], [5.533108, 52.510435], [5.533066, 52.509957], [5.533048, 52.509778], [5.533007, 52.509419], [5.532947, 52.508972], [5.532892, 52.508615], [5.532831, 52.508257], [5.532752, 52.507838], [5.53264, 52.507304], [5.532579, 52.50704], [5.532492, 52.506684], [5.532399, 52.506329], [5.532335, 52.506094], [5.532233, 52.50574], [5.532125, 52.505386], [5.532049, 52.505151], [5.531931, 52.504799], [5.53187, 52.504623], [5.531743, 52.504272], [5.531582, 52.503871], [5.531253, 52.503088], [5.531178, 52.502915], [5.531023, 52.502568], [5.530816, 52.50212], [5.530653, 52.501774], [5.530484, 52.50143], [5.530396, 52.501258], [5.530267, 52.501018], [5.530075, 52.500678], [5.529631, 52.499909], [5.529224, 52.499226], [5.529011, 52.498891], [5.528791, 52.498557], [5.528636, 52.498328], [5.528404, 52.497998], [5.528168, 52.497669], [5.527982, 52.497414], [5.527738, 52.497086], [5.527487, 52.496761], [5.527326, 52.496558], [5.527068, 52.496235], [5.526935, 52.496075], [5.526643, 52.495726], [5.52637, 52.495407], [5.526091, 52.49509], [5.525822, 52.49479], [5.525679, 52.494633], [5.525386, 52.494321], [5.524939, 52.493856], [5.524746, 52.49366], [5.52444, 52.493352], [5.524128, 52.493047], [5.523906, 52.492835], [5.523422, 52.492383], [5.522934, 52.491941], [5.522599, 52.491645], [5.522209, 52.49131], [5.522036, 52.491165], [5.521688, 52.490875], [5.521512, 52.490731], [5.521335, 52.490587], [5.520917, 52.490255], [5.520555, 52.489972], [5.520288, 52.489768]], “type”: “LineString”}, “properties”: {“endDateTime”: “2023-10-01T18:37:48Z”, “HgvCharge”: true, “id”: 600138410, “length”: 6422.33665139401, “name”: “Rijksweg A6”, “startDateTime”: “2023-10-01T18:29:57Z”}, “type”: “Feature”}, {“geometry”: {“coordinates”: [[5.520288, 52.489768], [5.519917, 52.489489], [5.519351, 52.489074], [5.519048, 52.488857], [5.518662, 52.488585], [5.518467, 52.48845], [5.517962, 52.488108], [5.517362, 52.487712], [5.516993, 52.487475], [5.516482, 52.487153], [5.51586, 52.48677], [5.515506, 52.486556], [5.515097, 52.486314], [5.514668, 52.486068], [5.514237, 52.485823], [5.513832, 52.485599], [5.513612, 52.485479], [5.513127, 52.485219], [5.512682, 52.484984], [5.512266, 52.484767], [5.51204, 52.484652], [5.511456, 52.484359], [5.510768, 52.484022], [5.510273, 52.483783], [5.509809, 52.483562],

[5.509173, 52.483264], [5.508703, 52.483048], [5.50823, 52.482834], [5.502673, 52.480344], [5.501991, 52.480035]], “type”: “LineString”}, “properties”: {“endDateTime”: “2023-10-01T18:40:01Z”, “HgvCharge”: true, “id”: 600137671, “length”: 1654.47009280056, “name”: “Rijksweg A6”, “startDateTime”: “2023-10-01T18:37:48Z”}, “type”: “Feature”}, {“geometry”: {“coordinates”: [[5.501991, 52.480035], [5.501585, 52.479929], [5.501217, 52.479779], [5.50097, 52.479681], [5.500845, 52.479633], [5.500781, 52.479611], [5.50066, 52.479573]], “type”: “LineString”}, “properties”: {“endDateTime”: “2023-10-01T18:40:11Z”, “HgvCharge”: true, “id”: 325398053, “length”: 104.352482884618, “name”: “Lelystad 10”, “startDateTime”: “2023-10-01T18:40:01Z”}, “type”: “Feature”}, {“geometry”: {“coordinates”: [[5.50066, 52.479573], [5.500592, 52.479555], [5.500523, 52.479539], [5.500417, 52.479524], [5.500345, 52.479518], [5.500231, 52.479515], [5.500159, 52.479519], [5.500086, 52.479528], [5.500015, 52.47954], [5.499936, 52.47956], [5.499873, 52.479582], [5.499812, 52.479608], [5.49975, 52.47964], [5.499699, 52.479672], [5.499653, 52.479707], [5.499615, 52.479741], [5.49958, 52.47978], [5.499548, 52.479831], [5.499529, 52.479875], [5.499514, 52.479936], [5.499511, 52.47998], [5.499522, 52.480049], [5.499539, 52.480093], [5.499564, 52.480141], [5.499591, 52.480182], [5.499624, 52.480222], [5.499677, 52.480269], [5.499722, 52.480304], [5.499772, 52.480337], [5.499975, 52.480457], [5.500137, 52.480549], [5.500393, 52.480689], [5.501099, 52.481066], [5.501394, 52.481235]], “type”: “LineString”}, “properties”: {“endDateTime”: “2023-10-01T18:40:57Z”, “HgvCharge”: true, “id”: 325398052, “length”: 302.123917041746, “name”: “Lelystad 10”, “startDateTime”: “2023-10-01T18:40:11Z”}, “type”: “Feature”}, {“geometry”: {“coordinates”: [[5.51698320890958, 52.5546344903841], [5.51728954643585, 52.5544625626869], [5.51882123406997, 52.5535456035908], [5.51883301628189, 52.5534739653544], [5.518809451858, 52.553402327001], [5.51501557941063, 52.5510955095069], [5.51467038896436, 52.5507907931857], [5.51445830913795, 52.5505185501746], [5.51437583365004, 52.550303620289], [5.51445830913795, 52.5502104836784], [5.51461147790036, 52.5500743605839], [5.51436405143667, 52.5498379352587], [5.51425801152351, 52.5496230020397], [5.51412840718635, 52.5493149292565], [5.51393989178527, 52.5491931324434], [5.51402236727316, 52.5490426770901], [5.51429335816215, 52.5489925251911], [5.51454078462584, 52.5490570061937], [5.52214031173258, 52.5485053323314], [5.52717133512473, 52.5481471038931], [5.52743054380034, 52.5480897862627], [5.52759549477616, 52.5479679860495], [5.52780757460255, 52.5479751507773], [5.52801965442896, 52.5480467979905], [5.52805500106624, 52.5482044214487], [5.5279371789411, 52.5483405503428], [5.5279136145158, 52.5484336909223], [5.52794896115446, 52.5485196666659]], “type”: “LineString”}, “properties”: {“endDateTime”: “2023-10-01T18:42:33Z”, “HgvCharge”: false, “startDateTime”: “2023-10-01T18:40:57Z”}, “type”: “Feature”}], “type”: “FeatureCollection”}

This is a publication of:

The Ministry of Infrastructure and Water Management

Postbus 20901
2500 EX The Hague

June 2025