



Ministry of Infrastructure
and Water Management

Message Specification Document candidate- (E)ETS- provider Temporary Toll

Annex to Practical Information Accreditation and
Toll services

01-07-2024

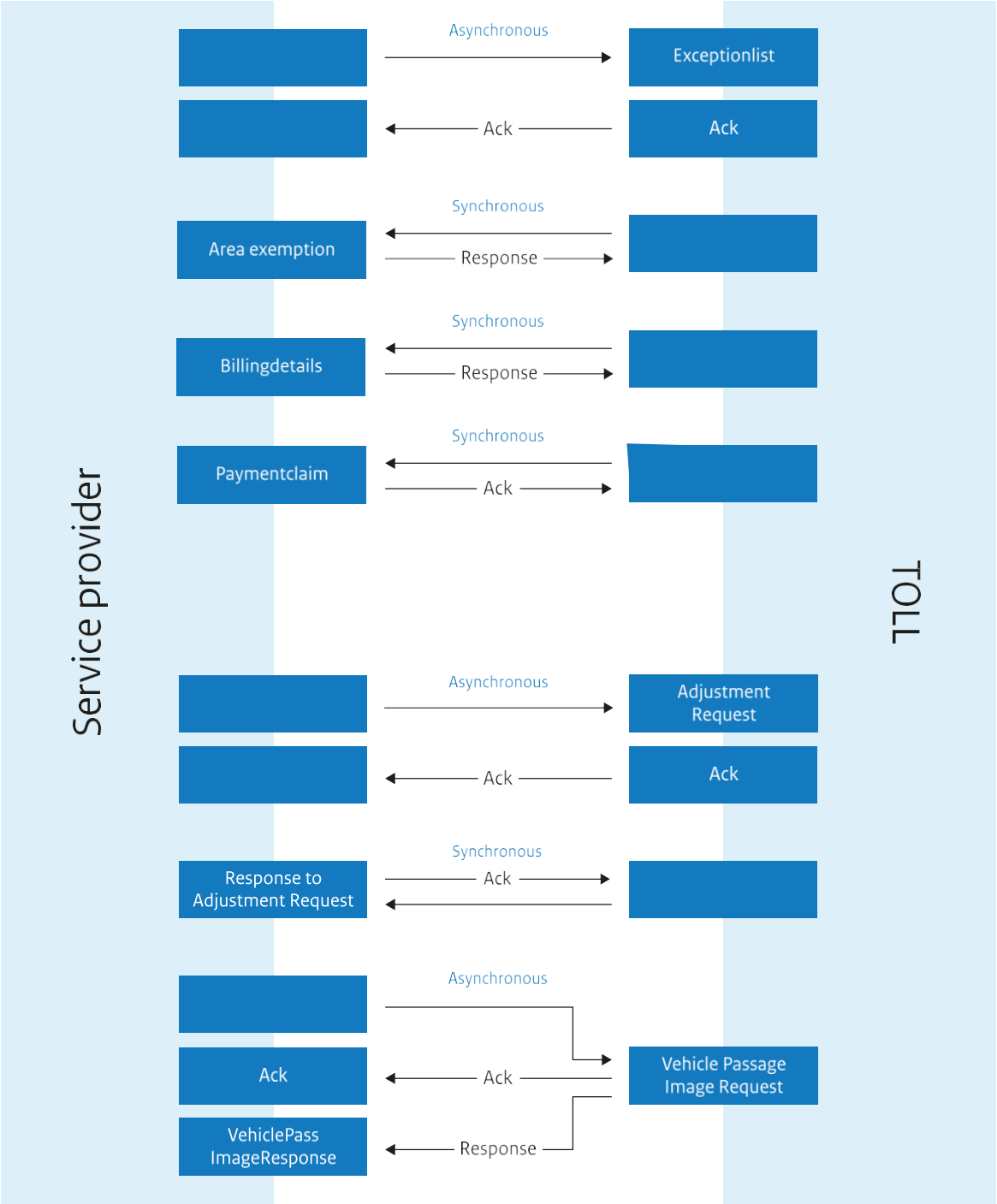
Table of contents

1	Introduction	3
1.1	Scope	4
1.2	Document management	4
1.3	Target audience	4
1.4	Norms	4
2	Message specification	5
2.1	General	5
2.2	Exceptionlist	5
2.2.1	Whitelist	5
2.2.2	Blacklist	8
2.3	Billingdetails	10
2.4	Paymentclaim	12
2.5	Ack return message	13
2.6	Adjustment Request	14
2.7	Response to Adjustment Request	17
2.8	Vehicle Passage Image Request	18
2.9	Area exemption	20
2.10	Countrycode	21

1 Introduction

The system for TOLL has a number of interfaces with the service provider. This interface document mentions the interfaces.

In the figure below the interfaces are mentioned.



1.1 Scope

This document mentions the interface with the service providers.

These service providers are: EETS-provider (European Electronic Toll Service), ETS-provider (Electronic Toll Service) and MSP (Main Service Provider).

In this document you will find the various services, the request and reply parameters.

1.2 Document management

This document is managed by Toll provider.

1.3 Target audience

This document is for all involved parties concerning specifying, developing, testing, implementing and managing of the interfaces. This for the TOLL-system.

1.4 Norms

The interface has to meet the following norms:

- W3C WS-* norm
- ISO12855:2015 and the CEN/TS 16986:2016/19
- X509 v3 certificates
- "XML Signature Syntax and Processing Version 2.0", www.w3.org/2000/09/xmldsig
- "Canonicalization algorithm", xml-exc-c14n, 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

2 Message specification

2.1 General

In all the messages identifying numbers are used for the Toll Services Provider (TSP) and for the Toll Charger (TC). For every tolldomain a different toll charger ID is defined.

2.2 Exceptionlist

Message for reporting changes of the customer list of a TOLL service provider. This message is defined in ISO12855:2015 and CEN/TS 16986 as exceptionListADU.

2.2.1 Whitelist

For every new contract or contract where suspension ends, an exceptionlist (whitelist) will be sent to the TOLL charger. Only one contract per message. Every EETS area has its own TOLL charger. The id of the TOLL charger has to be provided in the informationRecipientID.

Directly after receiving the Whitelist from the TSP, the TC will create the contract in its administration and send an Ack return message to the TSP. The most recent date of exceptionValidityStart, entryValidityStart and system date/time is the startdate/time of the contract.

This startdate/time will also be provided in the ACK message sent to TSP right after creating the contract at TC side. Even if the ACK message will not be received at once at TSP side, the value of the apduDate will not change.

If the exceptionvalidity or entryvalidity are in the future, then the apduDate of the ACK will not match the registered contract start date at TC side (and not at TSP side either). But then, the newest of exceptionvalidity and entryvalidity is the starting date of the contract at both TSP and TC sides.

Every new ExceptionList will have an ascending subsequent exceptionListVersion. After reaching number 255, numbering will start over from 0.

A new ExceptionList *for the same tolldomain* may not be sent unless the previous ExceptionList has been technically acknowledged by the TC.

Request message

Data item	Type and length	Possible values
infoExchange		
infoExchangeContent		
apci		
aidIdentifier	nonNegativeInteger 19	0: ISO12855:2015
apduOriginator		Service Provider
countryCode	String 15	See paragraph 2.9
providerIdentifier	nonNegativeInteger 10	
informationSenderID		Service Provider
countryCode	String 15	See paragraph 2.9
providerIdentifier	nonNegativeInteger 10	

Data item	Type and length	Possible values
informationRecipientID		TOLL charger
countryCode	String 15	See paragraph 2.9
providerIdentifier	nonNegativeInteger 10	
apduIdentifier	nonNegativeInteger 19	
apduDate	dateTime	Iso8601DateTimeUtc
adus		
exceptionListADU		
exceptionListVersion	nonNegativeInteger 3	0 till 255 ascending. After 255 start with 0
exceptionListType	nonNegativeInteger 3	2: Whitelist
exceptionValidityStart	dateTime	Iso8601DateTimeUtc
exceptionListEntries		
exceptionListEntry		
userId		
licencePlateNumber		
countryCode	String 15	See paragraph 2.9
alphabetIndicator	string	13: UTF-8 notation
licencePlateNumber	hexBinary 15	For German and Austrian licenceplates the landkreis or districtcode is separated from the rest of the licenceplatenumber by a hyphen sign. For example a licenceplate NOH 777 will be provided as NOH-777.
statusType	nonNegativeInteger 3	3: noLimits
reasonCode	nonNegativeInteger 3	8: user in whitelist
entryValidityStart	dateTime	Iso8601DateTimeUtc
vehicleParameters		
dateAndTime	dateTime	
vehicleClass	nonNegativeInteger 1	1: <= 3500 kg 2: > 3500 kg
actionRequested	nonNegativeInteger 3	3: accept this OBE

If the message is not correct, then a technical error will be sent to service provider:
 Apdureasoncode = 3 (APDU rejected) en AduReasonCode = 13000 (semanticError).

The following functional checks will be done on the common part of the message:

Check	Error	Number
The field aidIdentifier has the value 0, otherwise error message.	APDU rejected because Protocol Version not supported	Apdureasoncode = 206
Information recipient has a valid value, otherwise error message.	APDU rejected because Information Recipient not known (or no valid contract exists)	Apdureasoncode = 208
A valid contract exists with the delivering service provider, otherwise error message.	APDU rejected because Apdu Originator not known or no valid contract exists	Apdureasoncode = 207
The field adus contains 'exception- ListADUs', otherwise error message.	APDU rejected The specified ADU is invalid	Apdureasoncode = 3 AduReasonCode = 0
The field exceptionListVersion contains a value >= 0 and <= 255, otherwise error message.)	APDU rejected The exception list version has been rejected	Apdureasoncode = 3 AduReasonCode = 400
The field exceptionListType contains a value 2 (whitelist), otherwise error message.	APDU rejected The type of exception list has been rejected	Apdureasoncode = 3 AduReasonCode = 401

The following functional checks will be done on the specific part of the message:

Check	Error	Number
The field countryCode contains a value from the ISO 3166-1-Alpha-2, otherwise error message.	APDU accepted The user ID in exception list has been rejected	Apdureasoncode = 2 AduReasonCode = 402 IssueText = "countryCode"
The field alphabetIndicator contains the value 13, otherwise error message.	APDU accepted The user ID in exception list has been rejected	Apdureasoncode = 2 AduReasonCode = 402 IssueText = "alphabetIndicator"
The field reasonCode contains the value 8, otherwise error message.	APDU accepted The reason code in exception list has been rejected	Apdureasoncode = 2 AduReasonCode = 404
The field actionRequested contains the value 3, otherwise error message.	APDU accepted The actionRequested in exception list has been rejected	Apdureasoncode = 2 AduReasonCode = 10406
There is no actual exemption registered at TC side for the licenceplatenumber, otherwise error message.	APDU accepted The user ID in exception list has been rejected	Apdureasoncode = 2 AduReasonCode = 402 IssueText = "licencePlateNumber"
A German or Austrian licenceplate must contain exactly one "-"	APDU accepted The user ID in exception list has been rejected	Apdureasoncode = 2 AduReasonCode = 402 IssueText = "licencePlateformat"

Licencesplates other than German and Austrian may only contain letters and numbers.	APDU accepted The user ID in exception list has been rejected	Apdureasoncode = 2 AduReasonCode = 402 IssueText = "licencePlateformat"
---	--	---

2.2.2 Blacklist

For every contract where suspension starts, or every contract which ends, an exceptionlist (blacklist) will be sent to the TOLL charger. Only one contract per message. Every EETS area has its own TOLL charger. The moment this contract is registered at the TOLL charger, the contract has become invalid.

Directly after receiving the Blacklist from the TSP, the TC will end or suspend the contract in its administration and send an Ack return message to the TSP. The most recent date of exceptionValidityStart, entryValidityStart and system date/ time is the startdate/time of the contract.

This startdate/time will also be provided in the ACK message sent to TSP right after ending or suspending the contract at TC side. Even if the ACK message will not be received at once at TSP side, the value of the apduDate will not change.

Every new ExceptionList will have an ascending subsequent exceptionListVersion. After reaching number 255, numbering will start over from 0.

A new ExceptionList for the same licenceplate, countrycode and tolldomain may not be sent unless the previous ExceptionList has been technically acknowledged by the TC.

Request message

Data item	Type and length	Possible values
infoExchange		
infoExchangeContent		
apci		
aidIdentifier	nonNegativeInteger 19	0: ISO12855:2015
apduOriginator		Service Provider
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
informationSenderID		Service Provider
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
informationRecipientID		TOLL charger
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
apduIdentifier	nonNegativeInteger19	
apduDate	dateTime	Iso8601DateTimeUtc
adus		
exceptionListADU		
exceptionListVersion	nonNegativeInteger 3	0 till 255 ascending. After 255 start with 0
exceptionListType	nonNegativeInteger 3	1: Blacklist
exceptionValidityStart	dateTime	Iso8601DateTimeUtc
exceptionListEntries		
exceptionListEntry		

Data item	Type and length	Possible values
userId		
licencePlateNumber		
countryCode	String 15	See paragraph 2.10
alphabetIndicator	string	13: UTF-8 notation
licencePlateNumber	hexBinary 15	For German and Austrian licenceplates the landkreis or districtcode is separated from the rest of the licenceplatenumber by a hyphen sign. For example a licenceplate NOH 777 will be provided as NOH-777.
statusType	nonNegativeInteger 3	0: user is blocked
reasonCode	nonNegativeInteger 3	0: reason not to be disclosed
entryValidityStart	dateTime	Iso8601DateTimeUtc
vehicleParameters		
dateAndTime	dateTime	
vehicleClass	nonNegativeInteger 1	1: <= 3500 kg 2: > 3500 kg
actionRequested	nonNegativeInteger 3	1: reject this OBE (ending) 2: invalidate this OBE (suspension)

If the message is not correct, then a technical error will be sent to service provider:
 Apdureasoncode = 3 (APDU rejected) en AduReasonCode = 13000 (semanticError).

The following functional checks will be done on the common part of the message:

Check	Error	Number
The field aidIdentifier has the value 0, otherwise error message.	APDU rejected because Protocol Version not supported	Apdureasoncode = 206
Information recipient has a valid value, otherwise error message.	APDU rejected because Information Recipient not known (or no valid contract exists)	Apdureasoncode = 208
A valid contract exists with the delivering service provider, otherwise error message.	APDU rejected because Apdu Originator not known or no valid contract exists	Apdureasoncode = 207
The field adus contains 'exception- ListADUs', otherwise error message.	APDU rejected The specified ADU is invalid	Apdureasoncode = 3 AduReasonCode = 0
The field exceptionListVersion contains a value >= 0 and <= 255, otherwise error message.)	APDU rejected The exception list version has been rejected	Apdureasoncode = 3 AduReasonCode = 400
The field exceptionListType contains a value 1 (whitelist), otherwise error message.	APDU rejected The type of exception list has been rejected	Apdureasoncode = 3 AduReasonCode = 401

The following functional checks will be done on the specific part of the message:

Check	Error	Number
The field countryCode contains a value from the ISO 3166-1-Alpha-2, otherwise error message.	APDU accepted The user ID in exception list has been rejected	Apdureasoncode = 2 AduReasonCode = 402 IssueText = "countryCode"
The field alphabetIndicator contains the value 13, otherwise error message.	APDU accepted The user ID in exception list has been rejected	Apdureasoncode = 2 AduReasonCode = 402 IssueText = "alphabetIndicator"
The field reasonCode contains the value 0, otherwise error message.	APDU accepted The reason code in exception list has been rejected	Apdureasoncode = 2 AduReasonCode = 404
The field actionRequested contains the value 1 or 2, otherwise error message.	APDU accepted The actionRequested in exception list has been rejected	Apdureasoncode = 2 AduReasonCode = 10406
There is a (future) actual contract or suspended contract registered for the licenceplatenumber, otherwise error message.	APDU accepted The user ID in exception list has been rejected	Apdureasoncode = 2 AduReasonCode = 402 IssueText = "licencePlateNumber"
A German or Austrian licenceplate must contain exactly one "-"	APDU accepted The user ID in exception list has been rejected	Apdureasoncode = 2 AduReasonCode = 402 IssueText = "licencePlateformat"
Licenceplates other than German and Austrian may only contain letters and numbers.	APDU accepted The user ID in exception list has been rejected	Apdureasoncode = 2 AduReasonCode = 402 IssueText = "licencePlateformat"

2.3 Billingdetails

The billingdetails is a service that runs with the service provider. For this service we offer the following message.

Message for reporting the allocated vehicle passages to the service provider. This message is defined in ISO12855:2015 and CEN/TS 16986 as billingDetailsADU. For every vehicle passage, a BillingDetails message is sent.

For every vehicle passage, a BillingDetails message is directly sent to the service provider. The BillingDetailsNum contains the unique identifier of the BillingDetails. The paymentReference field will be used to combine the BillingDetails into a single PaymentClaim.

When an Adjustment Request has been accepted by the TOLL charger, the BillingDetails has to be

revoked. A new BillingDetails will be sent to the service provider with actionCode = 1 (revoke). In that

case the billing-

Details will be populated the same as the original one, except for apduIdentifier (new value), actionCode (1) and paymentReference (new value, can be the same if it is on the same day).

Additionally a new BillingDetails can be sent with correct values, in case of adjusted vehicleClass. Then, the field relatedBillingDetails will contain the value of the field billingDetailsNum of the original BillingDetails. It is further

populated as a new BillingDetails.

The BillingDetails message can only be accepted (apduAckCode = 2) in the Ack return message. Possible complaints must be sent through an Adjustment Request.

Data item	Type en length	Possible values
infoExchange		
infoExchangeContent		
apci		
aidIdentifier	nonNegativeInteger 19	
apduOriginator		TOLL charger
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
informationSenderID		TOLL charger
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
informationRecipientID		Service Provider
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
apduIdentifier	nonNegativeInteger 19	
apduDate	dateTime	
adus		
billingDetailsADU		
billingDetailsId		
issuerId		TOLL charger
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
billingDetailsNum	nonNegativeInteger 19	
tollContext		TOLL charger
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
userId		
licencePlateNumber		
countryCode	String 15	See paragraph 2.10
alphabetIndicator	string	13: UTF-8 notation

Data item	Type en length	Possible values
licencePlateNumber	hexBinary 15	For German and Austrian licenceplates the landkreis or districtcode is separated from the rest of the licenceplate number by a hyphen sign. For example a licenceplate NOH 777 will be provided as NOH-777.
relatedBillingDetails	nonNegativeInteger 19	apduIdentifier of related BillingDetails in case of revocation
billingDetailsAmount		
paymentFeeAmount	integer 19	Fee in euro-cent
paymentFeeUnit	hexBinary 10	2978: Euro-Cent
usageDetails		
contextName	string 10	EETS area and driving direction: 'BBV-L', 'BBV-R', 'ViA15-L' or 'ViA15-R'
appliedUserClass	string 15	'Not relevant'
perDeclaredVehicleClasses		
declaredVehicleClass	nonNegativeInteger 2	
perUsedTimeClasses		
appliedTimeClass	String 15	'Not relevant'
usageList		
usageListEntry		
forSectionedRoads		
howManyTimes	nonNegativeInteger 3	1
listOfSections		
Section		
tollEventTime	dateTime	Date and time of the vehicle passage
actionCode	nonNegativeInteger 3	0: normal sending 1: revoke (in case of accepted adjustment request)
paymentReference	String 32	

2.4 Paymentclaim

The paymentclaim is a service that runs with the service provider. For this service we offer the following message.

Message for reporting a paymentrequest to the service provider, concerning the allocated vehicle passages with a certain paymentreference per EETS-area per service provider. This message is defined in ISO12855:2015 and CEN/TS 16986 as paymentClaimADU.

Every day, a paymentclaim message is sent to the service provider, which contains the sum of all the BillingDetails with the same paymentReference as mentioned in the paymentClaim. If a BillingDetails has an actionCode 1 (revoked), the paymentFeeAmount will be subtracted from the total paymentFeeAmount of the PaymentClaim.

The PaymentClaim message can only be accepted (apduAckCode = 2) in the Ack return message. Possible complaints cannot be handled through the system, but must be handled manually.

Data item	Type en length	Possible values
infoExchange		
infoExchangeContent		
apci		
aidIdentifier	nonNegativeInteger 19	0: ISO12855:2015
apduOriginator		TOLL charger
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
informationSenderID		TOLL charger
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	

Data item	Type en length	Possible values
informationRecipientID		Service Provider
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
apduIdentifier	nonNegativeInteger 19	
apduDate	dateTime	
adus		
paymentClaimADU		
paymentClaimId	nonNegativeInteger 19	
startDateTime	dateTime	
endDateTime	dateTime	
paymentClaimAmount		
paymentFeeAmount	Integer 19	
paymentFeeUnit	hexBinary 10	2978: Euro-cent
paymentClaimStatus	nonNegativeInteger 3	0: firstVersion
typeOfFee	nonNegativeInteger 3	0: toll
actioncode	nonNegativeInteger 3	0: normal sending
paymentReference	String 32	paymentReference of the PaymentClaim

2.5 Ack return message

Message for the return message of the process of one or more received messages of the TOLL service provider. This message is defined in ISO12855:2015 and CEN/TS 16986 as AckADU.

Return message

Data item	Type and length	Possible values
infoExchange		
infoExchangeContent		
apci		
aidIdentifier	nonNegativeInteger 19	
apduOriginator		
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
informationSenderID		
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
informationRecipientID		
countryCode	String 15	See paragraph 2.10

Data item	Type and length	Possible values
providerIdentifier	nonNegativeInteger 10	
apduIdentifier	nonNegativeInteger 19	
apduDate	dateTime	
adus		
AckADU		
apduIdentifier	nonNegativeInteger 19	Value of apduIdentifier of original message
apduAckCode	nonNegativeInteger 3	Value of apduReasonCode of original message
issues		
issueADUStruct	nonNegativeInteger 19	0
issueCode	nonNegativeInteger 5	Value of aduReasonCode of original message
issueText	UTF8String 1024	

2.6 Adjustment Request

When the service provider does not agree with a specific BillingDetails, an adjustment request can be sent to the toll charger.

This message is defined in ISO12855:2022 as userComplaintADU. However, the 2022 version of the ISO standard is not (yet) supported by the Toll Charger. In order to support the process of Adjustment Request, the usage of the message userComplaint has been slightly altered in order to align with the supported 2015 standard.

When (the customer of) the service provider disagrees with a previously sent BillingDetails message, the service provider might file a complaint. For every complaint an Adjustment Request message has to be sent to the Toll Charger.

The Adjustment Request always refers to a BillingDetails message by using the BillingDetailsNum in 'relatedADUIdentifier'.

The service provider may send a maximum of only one Adjustment Request for a BillingDetails. The service provider may choose one of the following reasons for the

adjustment request:

1. wrongVehicleClass: when (according to the user and/or service provider) a wrong vehicle class (and therefore an incorrect payment fee amount) is billed
2. wrongUsageList: when (according to the user and/or service provider) the licenceplatenumber and/or country code is possibly wrongly recognized
3. wrongCharge – *will not be used*
4. chargeAlreadyProcessed: when (according to the user and/or service provider) a billing details is already charged by the Toll charger
5. otherReason: to indicate a reason not covered by the alternatives above, e.g. a stolen vehicle.

When using reason 'wrongVehicleClass' and the country code of the licenceplate <> NL, the service provider includes a copy of the registration certificate of the charged vehicle.

When using reason 'otherReason' and the country code of the licenceplate <> NL, the service provider includes a copy of the registration certificate of the charged vehicle. Other documents (i.e. police report in case of a stolen vehicle) may be included as well.

The Adjustment Request will be assessed by the toll charger. By using the Adjustment Request response the toll charger will inform the service provider whether the request was accepted or refused.

Data item	Type and length	Possible values
infoExchange		
infoExchangeContent		
apci		
aidIdentifier	nonNegativeInteger 19	
apduOriginator		
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
informationSenderID		
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
informationRecipientID		
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
apduIdentifier	nonNegativeInteger 19	
apduDate	dateTime	
userComplaintAdus		
userComplaintAdu		
aduIdentifier	nonNegativeInteger 19	
relatedId		
relatedAduType	nonNegativeInteger 3	
relatedAduIdentifier	nonNegativeInteger 19	
dateOfUserComplaint	dateTime	
userComplaintReason		
wrongVehicleClass		
billedVehicleClass		
localVehicleClassId	nonNegativeInteger 2	1: <= 3500 kg 2: > 3500 kg
claimedVehicleClass		
localVehicleClassId	nonNegativeInteger 2	1: <= 3500 kg 2: > 3500 kg
wrongUsageList	String 32	'wrongusagelist'
chargeAlreadyProcessed		
relatedId		
relatedAduType	nonNegativeInteger 3	
relatedAduIdentifier	nonNegativeInteger 19	

Data item	Type and length	Possible values
otherReason	String 32	
additionalUserComplaintInfo		0 - n
additionalUserComplaintInfoToBeSigned		
additionalUserComplaintInfoFold	nonNegativeInteger 19	
dateOfAdditionalUserComplaintInfo	dateTime	
infoData		
textItem	BASE64STRING	
binaryItem		
content	BASE64STRING	
mediaType	BASE64STRING	APPLICATION/PDF IMAGE/JPEG IMAGE/JPG IMAGE/PNG application/pdf image/jpeg image/jpg image/png

If the message is not correct, then a technical error will be sent to service provider:
 Apdureasoncode = 3 (APDU rejected) en AduReasonCode = 13000 (semanticError).

The following functional checks will be done on the common part of the message:

Check	Error	Number
The field aidIdentifier has the value 0, otherwise error message.	APDU rejected because Protocol Version not supported	Apdureasoncode = 206
Information recipient has a valid value, otherwise error message.	APDU rejected because Information Recipient not known (or no valid contract exists)	Apdureasoncode = 208
A valid contract exists with the delivering service provider, otherwise error message.	APDU rejected because Apdu Originator not known or no valid contract exists	Apdureasoncode = 207
The field adus contains 'userComplaintADUs', otherwise error message.	APDU rejected The specified ADU is invalid	Apdureasoncode = 3 AduReasonCode = 0

The following functional checks will be done on the specific part of the message:

Check	Error	Number
The billingDetailsNum in the field relatedADUIdentifier is known by the Toll charger	APDU accepted The billing details ID referred to in the user complaint ADU is not known by the TC	Apdureasoncode = 2 AduReasonCode = 11302
When using reason 'wrongVehicleClass' and countrycode of the vehicleplate <> NL, a document is attached.	APDU accepted Attachment is missing	Apdureasoncode = 2 AduReasonCode = 10001
When using reason 'otherReason' and countrycode <> NL, a document is attached.	APDU accepted Attachment is missing	Apdureasoncode = 2 AduReasonCode = 10001
Attachment is not larger than 15MB	APDU accepted Attachment too large	Apdureasoncode = 2 AduReasonCode = 10002
No more than 5 documents are attached.	APDU accepted Too many attachments	Apdureasoncode = 2 AduReasonCode = 10003

2.7 Response to Adjustment Request

With this message the Toll charger can inform the service provider whether the Adjustment Request has been accepted or refused.

This message is defined in ISO12855:2022 as userComplaintResponseADU. However, the 2022 version of the ISO standard is not (yet) supported by the Toll Charger. In order to support the process of Response to Adjustment Request, the usage of the message userComplaintResponse has been slightly altered in order to align with the supported 2015 standard.

The Response to Adjustment Request always refers to an Adjustment Request message by using the aduIdentifier in 'relatedUserComplaintId'.

Whether an Adjustment Request has been accepted or refused can be derived from the field userComplaintResponse.

Data item	Type and length	Possible values
infoExchange		
infoExchangeContent		
apci		
aidIdentifier	nonNegativeInteger 19	
apduOriginator		
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
informationSenderID		
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	

informationRecipientID		
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
apduIdentifier	nonNegativeInteger 19	

Data item	Type and length	Possible values
apduDate	dateTime	
userComplaintResponseAdus		
userComplaintResponseAdu		
aduIdentifier	nonNegativeInteger 19	
relatedUserComplaintId	nonNegativeInteger 19	The ADUIdentifier of the userComplaint
userComplaintResponse	nonNegativeInteger 3	0 – userComplaintAccepted 2 – userComplaintRefusedIncorrect-ClaimedLocalVehicleClass 3 – userComplaintRefusedIncorrect-ClaimedUsageList 5 – userComplaintRefusedCharge-AlreadyProcessed 10 – userComplaintRefusedOther-Reason

2.8 Vehicle Passage Image Request

With this message the service provider can request the accompanying photos of a specific vehicle passage. The billingDetailsNum of the passage should be provided (as 'relatedAduIdentifier') to receive the photos.

The url for this message is ../api/verstrekkenbeeldmateriaalaandienstaanbieder

Message to request the photos is described below.

Data item	Type and length	Possible values
infoExchange		
infoExchangeContent		
apci		
aidIdentifier	nonNegativeInteger 19	
apduOriginator		
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
informationSenderID		
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
informationRecipientID		
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
apduIdentifier	nonNegativeInteger 19	
apduDate	dateTime	
vehiclePassageImageRequestAdus		

Data item	Type and length	Possible values
vehiclePassageImageRequestAdu		
aduIdentifier	nonNegativeInteger 19	
relatedId		
relatedAduIdentifier	nonNegativeInteger 19	

If the message is not correct, then a technical error will be sent to service provider:
 Apdureasoncode = 3 (APDU rejected) en AduReasonCode = 13000 (semanticError).

The following functional checks will be done on the common part of the message:

Check	Error	Number
The field aidIdentifier has the value 0, otherwise error message.	APDU rejected because Protocol Version not supported	Apdureasoncode = 206
Information recipient has a valid value, otherwise error message.	APDU rejected because Information Recipient not known (or no valid contract exists)	Apdureasoncode = 208
A valid contract exists with the delivering service provider, otherwise error message.	APDU rejected because Apdu Originator not known or no valid contract exists	Apdureasoncode = 207
The field adus contains 'vehicle-PassageImageRequestADUs', otherwise error message.	APDU rejected The specified ADU is invalid	Apdureasoncode = 3 AduReasonCode = 0

The following functional checks will be done on specific part of the message:

Check	Error	Number
The field relatedAduIdentifier points to AduIdentifier of the request, otherwise error message.	APDU accepted vehiclePassageImageRequest - RelatedAduIdentifierUnknown	Apdureasoncode = 2 Adureasoncode = 11303
The images of the vehiclePassage are available, otherwise error message.	APDU accepted noVehiclePassageImagesAvailable	Apdureasoncode = 2 Adureasoncode = 11304

In the response message up to 5 images will be provided.

Data item	Type and length	Possible values
infoExchange		
infoExchangeContent		
apci		
aidIdentifier	nonNegativeInteger 19	
apduOriginator		
countryCode	String 15	See paragraph 2.10

Data item	Type and length	Possible values
providerIdentifier	nonNegativeInteger 10	
informationSenderID		
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
informationRecipientID		
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
apduIdentifier	nonNegativeInteger 19	
apduDate	dateTime	
vehiclePassImageResponseAdus		
vehiclePassImageResponseAdu		
aduIdentifier	nonNegativeInteger 19	
relatedId		
relatedAduIdentifier	nonNegativeInteger 19	
vehiclePassageImages		
vehiclePassageImage		
vehiclePassageImageID	String 38	
vehicleSide	String 1	F = Front; R = Rear
imageType	Base64	image/jpeg
imageRecordingType	String 1	I = Infrared; O = Overview; P = Patch image
image	Base64	

2.9 Area exemption

In case of emergency or maintenance around the Blankenburgtunnel for example, the BBV area will be TOLL-free for a period of time. In this period the service provider will not receive any BillingDetails. The message Area exemption will inform the service provider of the start or ending of this TOLL-free period, for a specific EETS area. In the message of the start of the area exemption, a reason for the exemption can be provided. With the stop of the area exemption, the end date/time is provided. The reason is then empty.

The message to the other service provider is as follows:

Data item	Type and length	Possible values
infoExchange		
infoExchangeContent		
apci		
aidIdentifier	nonNegativeInteger 19	
apduOriginator		
countryCode	String 15	See paragraph 2.10

Data item	Type and length	Possible values
providerIdentifier	nonNegativeInteger 10	
informationSenderID		
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
informationRecipientID		
countryCode	String 15	See paragraph 2.10
providerIdentifier	nonNegativeInteger 10	
apduIdentifier	nonNegativeInteger 19	
apduDate	dateTime	
areaExemptionNotificationAdus		
areaExemptionNotificationAdu		
adulIdentifier	nonNegativeInteger 19	
eetsAreaCode	String 10	BBV, ViA15
drivingDirection	String 5	LEFT, RIGHT
startDateTimeAreaExemption	dateTime	
endDateTimeAreaExemption	dateTime	
areaExemptionReason	String 255	

As this is a message just to inform the other service provider, only a technical acknowledgement message can be sent in return, there is no need for an Ack message.

2.10 Countrycode

The notation of the countrycode is a 10-digit bitstring. In this way the 2 characters of the countrycode are set in the message.

The possible values are combinations of:

(11000|10011|01110|10010|10000|10110|01011|00101|01100|11010|11110|01001|00111|00110|00011|01101|11010|10101|01010|10100|00001|11100|01111|11001|10111|10101|10001)

Where:

A = 11000

B = 10011

C = 01110

.....

Z = 10001

For example: 0011001001 stands for NL (Nederland)

This is a publication of:

Ministry of Infrastructure and Water Management
PO Box 20901 | 2500 EX The Hague | The Netherlands

www.rijksoverheid.nl/ienw

January 2024