

EasyGo+ and EETS DSRC transaction for tolling and enforcement

Enclosure C to Document 202 “Roadside and on-board equipment”

This copy of the document was published on www.easygo.com and is for information purposes only. It may change without further notice.

Table of contents

DOCUMENT REVISION HISTORY	3
ABBREVIATIONS	4
1 GENERAL.....	6
1.1 PURPOSE OF THE DOCUMENT	6
1.2 TOLLING AND ENFORCEMENT TRANSACTION.....	7
1.3 ENFORCEMENT TRANSACTION.....	8
1.4 SECURITY FEATURES	8
1.5 DATA SPECIFICATION	8
2 EASYGO+/ EETS TOLLING AND ENFORCEMENT TRANSACTION	9
2.1 DECISION FLOW.....	9
2.2 TOLLING AND ENFORCEMENT TRANSACTION - BIT-LEVEL SPECIFICATION	10
3 ENFORCEMENT TRANSACTION.....	21
3.1 ENFORCEMENT TRANSACTION - BIT-LEVEL SPECIFICATION	21
4 REFERENCES	25
4.1 STANDARDS AND EXTERNAL DOCUMENTS.....	25
4.2 EASYGO- DOCUMENTS	26

Document revision history

Version	Date	Author	Main changes
V0.1	23.02.2010	HHA	Initial compilation
V0.23	29.04.2010	HHA	Bit level description for security level 1 added
V0.24	02.07.2010	HHA	Ch. 1.2 and 1.3 presentation of AC_CR added
V0.25	19.10.2010	HHA	Editorial changes, release candidate (RC)
V1.0	18.11.2010	HHA	Release by WG2
V1.01	13.04.2011	HHA	Editorial changes
V1.50		HHA	Adaptation as enclosure to EasyGo document 202
V1.51	18.11.2011	HHA	Editorial changes
V1.52	15.01.2013	HHA	Ch. 1.3: no Get.Stamped (Get.Request only)
V1.53	11.04.2013	HHA	References updated
2.0	02.05.2013		Approved by steering committee
2.1	11.12.2017	HHA	Update for EETS, contract- and vehicle authenticator removed, editorial changes
2.2	22.02.2018	HHA	Release version
3.0	27.02.2018		Approved by Steering committee

Abbreviations

Abbreviation, Term	Description
AID	DSRC-Application Entity Identifier
AttrID	Attribute Identifier
BST	Beacon Service Table
CI	Contract Issuer = Toll Service Provider (TSP)
EasyGo+-OBE	Interoperable OBE
DES	Data Encryption Standard, ANSI X3.92
3DES	Triple Data Encryption Standard
DSRC	Dedicated Short Range Communication
EETS	European Electronic Toll Service
EFC	Electronic Fee Collection
EID	Element Identifier
FCS	Frame Check Sequence
HGV	Heavy Goods Vehicle
HV	Heavy Vehicle (> 3.5 tons)
IID	Invoker Identifier
KeyRef_CI	Key reference for TSP- (CI-) authenticator key
KeyRef_OP	Key reference for TC- (OP-) authenticator key
L1	Layer 1 of DSRC (Physical Layer)
L2	Layer 2 of DSRC (Data Link Layer)
L7	Layer 7 of DSRC (Application Layer)
LID	Link Identifier
LLC	Logical Link Control

Abbreviation, Term	Description
LPN	Licence Plate Number
MAC	Medium Access Control
MAC	Message Authentication Code
MMI	Man Machine Interface
OBE	On-Board Equipment
OBU	On-Board Unit
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
RSE	Roadside Equipment
UI	User Interface (= MMI)
VST	Vehicle Service Table

1 General

1.1 Purpose of the document

This specification provides the definition for EasyGo+ and EETS DSRC transactions according to EN15509 at EasyGo toll domains.

This specification applies for post payment transactions for communication between OBE and roadside tolling equipment and in case of multilane free flow systems for combined tolling and enforcement RSE.

In general, the transaction is based on the “pick what you like” idea, i.e. it is up to the toll charger which attribute must be read to allow correct charging. This chapter gives only an exemplary definition of the DSRC transactions, between an EasyGo+- or EETS OBE and the roadside equipment (RSE).

The table below provides an overview of the attributes involved in the data exchanges of DSRC transactions associated with the EasyGo+-OBE.

ATTRIBUTES (EID>0)	AttrID	Tolling & enforcement	Enforcement
EFC Context Mark	0	X	X
VehicleLicencePlateNumber	16	X	X
VehicleClass	17	X	X
VehicleDimensions	18	*)	*)
VehicleAxles	19	X	X
VehicleWeightLimits	20	*)	*)
VehicleSpecificCharacteristics	22	X	X
EquipmentOBUId	24	X	X
EquipmentStatus	26	X	
PaymentMeans (including PAN)	32	X	X
ReceiptData1	33	X	X
ReceiptData2	34	X	X

*) Attributes 18 and 20 are not used in this exemplary description.

1.2 Tolling and enforcement transaction

A tolling and enforcement transaction is performed for the purpose of charging and, if applicable, enforcing the due tolling fee.

The following table shows an example for such a transaction, a dedicated implementation has to be tailored to the toll chargers needs.

Phase	Roadside Equipment		On-board unit	Remarks
<i>Initialisation</i>	INITIALISATION.request (BST)	→		RSE periodically sends BST.
(BST – VST)		←	INITIALISATION.response (VST) <ul style="list-style-type: none"> EFC-ContextMark AC_CR-KeyReference RndOBE 	A newly arrived OBE answers with VST. According to the information in the VST, the RSE decides whether the OBE/Contract is acceptable or not. IF not the presentation phase is not started.
<i>Presentation</i>	GET_STAMPED.request AC_CR <ul style="list-style-type: none"> PaymentMeans, including PersonalAccountNumber (RndRSE, KeyRef_OP) GET.request <ul style="list-style-type: none"> EquipmentOBUId ReceiptData1 EquipmentStatus Vehicle data: <ul style="list-style-type: none"> VehicleClass VehicleAxles VehicleSpecificCharacteristics 	→		Read PaymentMeans (including Personal Account Number) and request the OBE to calculate the OP authenticator for the TC Read data that serves as a basis for calculating the fee (i.e. vehicleClass, VehicleAxles, VehicleSpecificCharacteristics) and data for diagnostics, auditing and controlling (i.e. Equipment OBU Id, Receipt Data1, EquipmentStatus). The Attribute VehicleLicencePlateNumber is read out in the receipt phase in order to provide a better distribution of data in uplink frames between the presentation and the uplink phases .
		←	GET_STAMPED.response <ul style="list-style-type: none"> Authenticator (Auth_CI) GET.response	OBE responds with the requested data, plus the CI Authenticator computed over Payment Means (including Personal Account Number)
<i>Receipt</i>	GET_STAMPED.request AC_CR <ul style="list-style-type: none"> PaymentMeans, including PersonalAccountNumber (RndRSE, KeyRef_CI) GET.request AC_CR <ul style="list-style-type: none"> Vehicle data: <ul style="list-style-type: none"> VehicleLicencePlateNumber SET.request AC_CR <ul style="list-style-type: none"> ReceiptData1 ReceiptData2 EquipmentStatus SET_MMI.request	→		Read Payment Means (including Personal Account Number) and request the OBE to calculate an authenticator for the TSP. Write new receipt (or entry ticket) to ReceiptData1. Copy old receipt to ReceiptData2 Write new status information and increment transaction counter. Signal the transaction's result via the OBE's HMI: OK, Warning or NOK
		←	GET_STAMPED.response <ul style="list-style-type: none"> Authenticator (Auth_OP) GET.response SET.response SET_MMI.response	
<i>Tracking</i>	ECHO.request	→		Track OBE by exchanging dummy information The usage of Echo is optional, at the discretion of the RSE, and may be repeated (used e.g. at enforcement equipment in multilane free flow systems).
And		←	ECHO.response	
<i>Closing</i>	EVENT_REPORT.request (Release)	→		RSE closes transaction and releases OBE

1.3 Enforcement transaction

An enforcement transaction is performed for enforcement-only purposes, e.g. used by mobile enforcement equipment.

Phase	Roadside Equipment		On-board unit	Remarks
<i>Initialisation</i> (BST – VST)	INITIALISATION.request (BST)	→		
		←	INITIALISATION.response (VST)	
<i>Presentation 1</i>	GET.request AC_CR <ul style="list-style-type: none"> • EquipmentOBUId • EquipmentStatus (transaction counter) • Vehicle data: <ul style="list-style-type: none"> - VehicleLicencePlateNumber - VehicleClass - VehicleAxles - VehicleSpecificCharacteristics 	→		
		←	GET.response	
<i>Presentation 2</i>	GET.request AC_CR <ul style="list-style-type: none"> • ReceiptData1 • ReceiptData2 	→		
		←	GET.response	
<i>Closing</i>	EVENT_REPORT.request (Release)	→		

1.4 Security features

The EasyGo+/ EETS DSRC transaction comprises the following security features:

- Transaction counter, increased by the RSE, allowing detection of transaction sequencing anomalies in the central system
- Authentication to the Service Provider (Contract Issuer), i.e. challenge-response of PaymentMeans data using the GET_STAMPED function with the CI Key
- Authentication to the Toll Charger, i.e. challenge-response of PaymentMeans data using the GET_STAMPED function with the Operator Key
- Protected access to the OBE's data, through the implementation of Access Credentials (Security level 1 acc. to EN15509).

1.5 Data specification

For data definition (attribute data) refer to the document 202-B [EasyGo-202-B].

2 EasyGo+/ EETS tolling and enforcement transaction

2.1 Decision flow

Each time the RSE receives a VST from an OBE it analyses the attribute EFC-ContextMark and if necessary, the data elements EquipmentClass+ManufacturerID, in order to decide which application to use. If those elements correspond to one of the entries in the RSE's Contract Issuer List (A list with entries for valid OBEs, where the data is derived a.o. from AIT information), the RSE will proceed with the associated application to perform a suitable transaction. In case of an EasyGo+/ EETS- OBE the Tolling and Enforcement Transaction (object of the present document) has to apply.

If the VST contains a list of (more than one) EFC-ContextMark, the first entry will be used that can be matched with the Contract Issuer List.

Access Credentials are used in the further transactional steps if the entry in the Contract Issuer List indicates to use security level 1, which is mandatory for EasyGo+- and EETS OBE.

The VST contains an ApplicationContextMark for security level 1 (AC_CR used) with:

EFC-ContextMark, AC_CR_Keyreference, RndOBU1 (in total 16 Bytes)

Access Credentials are used in the further transactional steps.

The first GET_STAMPED.request is performed with the suitable authenticator key for the operator (=TC) authentication as defined in the Contract Issuer List for that EFC-ContextMark.

Note: The SET_MMI command is used with a container type compliant to the OBE, which can be controlled by an appropriate entry in the Contract Issuer List, preferable the container type acc. to the last draft version of [EFC-API] (69h) shall be used.

If the PaymentMeans-ExpiryDate is older than RSE current station's date, the OBE shall be handled like an OBE with invalid PaymentMeans (SET_MMI.request: 4 beep code).

Value "0" of PaymentMeans-ExpiryDate means that there is no Expiry Date to be checked.

If the 15th bit of EquipmentStatus is set to 1 the OBE shall be handled as a "blacklisted OBE", if the Contract Issuer List indicates to do so (SET_MMI.request: 4 beep code).

The second GET_STAMPED.request is performed with the suitable authenticator key for the CI (=TSP) authentication as defined in the Contract Issuer List for that EFC-ContextMark.

The RndRse number is the same for both GET_STAMPED.request.

¹ The AC_CR_Keyreference and RndOBU data elements are defined as octet strings with a container Choice = 2 and a length indicator.

2.2 Tolling and enforcement transaction - Bit-level specification

The specification accounts for the complete frame content (excluding the zero-bit insertions) of the data exchanged, including protocol information related to DSRC-L1, -L2 and -L7, in order to ensure unambiguity.

Note: this bit-level specification has been written with the following constraints:
 EquipmentOBUID with length 4+1 Byte and VehicleLicencePlateNumber with length 13 Byte (LPN with 10 characters – minimal length). The bit-level changes accordingly, when these attributes have different length (see [EasyGo-202-B]). As the transaction acc. to EN15509 is based on the idea to read only attribute data needed for the tolling functionality in the specific tolling context, the following tables are only representing examples regarding attributes.

2.2.1 Initialisation

2.2.1.1 Initialisation request (BST)

Octet #	Attribute / Field	Bits in Octet b ₇ b ₀	Description
1	FLAG	0111 1110	Start Flag
2	Broadcast LID	1111 1111	Link address for broadcast
3	MAC control field	1010 0000	The frame contains a command LPDU
4	LLC control field	0000 0011	UI command
5	Fragmentation header	1xxx x001	No fragmentation. PDU # shall never be set to 0000 ₂ or 0001 ₂ .
6	BST SEQUENCE {	1000	INITIALISATION.request
	OPTION indicator	0	NonmandApplications not present.
	BeaconId.ManufacturerId INTEGER (0..65535)	000	Manufacturer identifier:1 (=Kapsch). See [AVI No]
7		0000 0000	Register at www.nen.nl/cen278 for value assignment.
8		0000 1	
	BeaconId.IndividualId INTEGER (0..2 ²⁷ -1)	000	27 bit ID available for manufacturer. Example: Id=1052
9		0000 0000	
10		0000 0100	
11		0001 1100	
12	Time TimeReal	0100 0001	32 bit UNIX System Time, the number of seconds passed
13		1100 1010	since 1st January 1970, 00:00 (UTC). Example: 1103790512 ₁₀
14		1000 0001	
15		1011 0000	
16	Profile INTEGER (0..127,...)	0000 0000	Profile (p=0 ₁₀ : 1,5 MHz sub-carrier, p=1 ₁₀ : 2,0 MHz sub-carrier). Example: No extension, Profile 0
17	MandApplications SEQUENCE (0..127,...) OF {	0000 0001	No extension, Number of mandApplications= 1
18	OPTION indicator	0	EID not present
	OPTION indicator	0	Parameter not present
	AID_DSRCApplicationEntityID }	00 0001	No extension. AID = 1 ₁₀ =EFC
19	ProfileList SEQUENCE (0..127,...) OF Profile }	0000 0000	No extension, number of profiles in list = 0.
20	FCS	xxxx xxxx	Frame check sequence
21		xxxx xxxx	
22	FLAG	0111 1110	End Flag

2.2.1.2 Private window request

Octet #	Attribute / Field	Bits in Octet b ₇ b ₀	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	0110 0000	Private window request
7	FCS	xxxx xxxx	Frame check sequence
8		xxxx xxxx	

Octet #	Attribute / Field	Bits in Octet b ₇ b ₀	Description
9	FLAG	0111 1110	End Flag

2.2.1.3 Private window allocation

Octet #	Attribute / Field	Bits in Octet b ₇ b ₀	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	0010 s000	Private window allocation
7	FCS	xxxx xxxx	Frame check sequence
8		xxxx xxxx	
9	FLAG	0111 1110	End Flag

2.2.1.4 Initialisation response (VST)

Note: the present VST is an example of a VST containing only one application and EFC-ContextMark. VSTs with more EFC-ContextMarks (within the L2 frame length limit) shall also be correctly interpreted by the RSE.

Octet #	Attribute / Field	Bits in Octet b ₇ b ₀	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	1100 0000	The frame contains a command LPDU
7	LLC control field	0000 0011	UI command
8	Fragmentation header	1xxx x001	No fragmentation. PDU # shall never be set to 0000 ₂ or 0001 ₂ .
9	VST SEQUENCE {	1001	INITIALISATION.response
	Fill BIT STRING (SIZE(4))	0000	Set to 0
10	Profile INTEGER (0..127,...)	0000 0000	No extension, profile p. Example : 0 ₁₀
11	Applications SEQUENCE (0..127,...) OF {	0000 0001	No extension, 1 application
12	OPTION indicator	1	EID present
	OPTION indicator	1	Parameter present
	AID DSRCApplicationEntityID	00 0001	No extension, AID = 1 (EFC)
13	EID	0000 0101	Uniquely associated with a context mark in the OBE. Example : 5 ₁₀
14	Parameter CONTAINER {	0000 0010	Choice 2 = Octet string
15		0000 0110	No extension, octet string length = 6 ₁₀
16	EFC-ContextMark SEQUENCE {		
	ContractProvider SEQUENCE {		
	CountryCode BIT STRING (SIZE(10))	0111 0100	As Specified in [EasyGo-202-B]
17		01	
	IssuerIdentifier INTEGER (0..16383) }	00 0000	As Specified in [EasyGo-202-B]
18		0000 0001	
19	TypeOfContract OCTET STRING (SIZE(2))	0000 0011	Type of contract. As Specified in [EasyGo-202-B]
20		0000 0000	

Octet #	Attribute / Field	Bits in Octet b ₇ b ₀	Description
21	ContextVersion INTEGER (0..127...) }}	0000 0001	No extension, context version. As Specified in [EasyGo-202-B]
A	CONTAINER	0000 0010	Choice 2 = Octet string
B	OCTET STRING	0000 0010	No extension, field length 2 ₁₀
C	AC_CR-Reference SEQUENCE { AC-MasterKeyRef Int1, AC_CR-Diversifier Int1}	0000 0001	AC_CR-Reference to, consisting of AC_CR-MasterKeyRef and AC_CR-Diversifier, used for the computation of AC_CRKey and AC_CR.
D	CONTAINER	0000 0010	Choice 2 = Octet string
F	OCTET STRING	0000 0100	No extension, field length 4 ₁₀
G	rndOBE Int4	0000 0000	Random Number (nonce) used together with AC_CRKey to calculate AC_CR. Example : 640 ₁₀
H		0000 0000	
I		0000 0010	
J		1000 0000	
22	ObeConfiguration SEQUENCE { OPTION indicator EquipmentClass INTEGER (0..32767)	1 000 0000 0000 0011	ObeStatus present Example : 3 ₁₀
23			
24	ManufacturerId INTEGER (0..65535)	0000 0000 0000 0001	Manufacturer identifier. See [AVI No] Register at www.nen.nl/cen278 for value assignment. Kapsch = 1 ₁₀ .
25			
26	ObeStatus INTEGER(0..65535)	0000 0011 0000 0000	Example : 768 ₁₀
27			
28	FCS	xxxx xxxx	Frame check sequence
29		xxxx xxxx	
30	FLAG	0111 1110	End Flag

2.2.2 Presentation

2.2.2.1 Presentation request

Octet #	Attribute / Field	Bits in Octet b ₇ b ₀	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	1010 s000	The frame contains a command LPDU
7	LLC control field	n111 0111	Polled ACn command, n bit
8	Fragmentation header	1xxx x001	No fragmentation. First service of chain.
9	GET_STAMPED.requestSEQUENCE {	0000 0101 0000 1101	ACTION.request (GET Stamped, AccessCredential not present, ActionParameter present, IID not present and Reply expected) or alternatively: ACTION.request (GET Stamped, AccessCredentials, ActionParameter present, IID not present and Reply expected)
10	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBE. Example : 5 ₁₀
11	ActionType INTEGER(0..127,...)	0000 0000	No extension, GET_STAMPED.request = 0
A	AccessCredential OCTET STRING {	0000 0100	No extension, octet string length = 4 ₁₀
B	AC_CR	0000 0100	Access credential calculated by RSE using RndOBE and the
C		1001 0100	Access Credential Key AC_CRKey. Example: AC_CR(0) =
D		1111 1000	04 94 F8 97'H
E		1001 0111	
12	ActionParameter CONTAINER {	0001 0001	No extension, Choice 17 ₁₀ = GetStampedRq
13	AttributeIdList SEQUENCE (0..127,...) OF { INTEGER (0..127,...) AttributeId {	0000 0001	No extension, number of attribute IDs = 1
14	PaymentMeans } }	0010 0000	AttributeId = 32 ₁₀ = PaymentMeans
15	Nonce OCTET STRING {	0000 0100	No extension, octet string length = 4 ₁₀
16	RndRSE	ffff ffff	Random number from RSE, containing SessionTime, needed to calculate the Authenticator
17		ffff ffff	
18		ffff ffff	
19		ffff ffff	
20	KeyRef } }	0111 0011	KeyRef_OP (e.g. =115 ₁₀)
21	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as before (concatenation).
22	GET.request SEQUENCE {	0110	GET.request
	OPTION indicator	0	AccessCredential not present or alternatively:
		1	AccessCredential present
	OPTION indicator	0	IID not present
	OPTION indicator	1	AttributeIdList present

Octet #	Attribute / Field	Bits in Octet b ₇ b ₀	Description
	Fill BIT STRING(SIZE(1))	0	Set to 0
23	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBE. Example : 5 ₁₀
A	AccessCredential OCTET STRING {	0000 0100	No extension, octet string length = 4 ₁₀
B	AC_CR	0000 0100	Access credential calculated by RSE using RndOBE and the
C		1001 0100	Access Credential Key AC_CRKey. Example: AC_CR(0) =
D		1111 1000	04 94 F8 97'H
E	}	1001 0111	
24	AttributeIdList SEQUENCE (0..127,...) OF { INTEGER (0..127,...) AttributeId {	0000 0110	No extension, number of attribute Ids = 6 ₁₀
25	VehicleClass	0001 0001	AttributeId = 17 ₁₀ = VehicleClass
26	VehicleAxles	0001 0011	AttributeId = 19 ₁₀ = VehicleAxles
27	VehicleSpecificCharacteristics	0001 0110	AttributeId = 22 ₁₀ = VehicleSpecificCharacteristics
28	EquipmentOBUId	0001 1000	AttributeId = 24 ₁₀ = EquipmentOBUId
29	EquipmentStatus	0001 1010	AttributeId = 26 ₁₀ = EquipmentStatus
30	ReceiptData1 } } }	0010 0001	AttributeId = 33 ₁₀ = ReceiptData1
31	FCS	xxxx xxxx	Frame check sequence
32		xxxx xxxx	
33	FLAG	0111 1110	End Flag

2.2.2.2 Presentation response

Octet #	Attribute / Field	Bits in Octet b ₇ b ₀	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	1101 0000	The frame contains a response LPDU
7	LLC control field	n111 0111	Response available, Acn command n bit
8	LLC status field	0000 0000	Response available and command accepted
9	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as in the corresponding request. .
10	GET_STAMPED.response SEQUENCE {	0001 0100	ACTION.response (Get Stamped rs)
11	EID INTEGER (0..127,...)	0000 0101	Uniquely associated with a context mark in the OBE. Example : 5 ₁₀
12	ResponseParameter CONTAINER {	0001 0010	No extension. Choice 18 ₁₀ = GetStampedRs
13	AttributeList SEQUENCE (0..127,...) OF {	0000 0001	No extension, number of attributes: 1
14	Attributes SEQUENCE { AttributeId	0010 0000	PaymentMeans = 32 ₁₀
15	Attribute Value CONTAINER {	0100 0000	Container Choice: 64 ₁₀ = PaymentMeans
16	PersonalAccountNumber	xxxx xxxx	PersonalAccountNumber
17		xxxx xxxx	
18		xxxx xxxx	
19		xxxx xxxx	
20		xxxx xxxx	
21		xxxx xxxx	
22		xxxx xxxx	
23		xxxx xxxx	
24		xxxx xxxx	
25		xxxx xxxx	
26	PaymentMeansExpiryDate	0001 1110	DateCompact. Example : 2005-03-01
27		0110 0001	
28	PaymentMeansUsageControl	0000 0000	Example : Not specified 0 ₁₀
29		0000 0000	
30	Authenticator OCTET STRING {	0000 0100	No extension, octet string size = 4 ₁₀
31	Authenticator	xxxx xxxx	Authenticator over AttributeList (containing PaymentMeans) and RndRSE (containing SessionTime) calculated using AuKey
32		xxxx xxxx	
33		xxxx xxxx	
34		xxxx xxxx	
35	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as in the corresponding request.
36	GET.response SEQUENCE	0111 0100	GET.response
37	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBE. Example : 5 ₁₀
38	AttributeList SEQUENCE (0..127,...) OF {	0000 0110	No extension, 6 attributes in list.
39	AttributeId INTEGER(0..127,...)	0001 0001	AttributeId = 17 ₁₀ = VehicleClass
40	Attribute Value CONTAINER {	0011 0001	Container choice = 49 ₁₀
41	VehicleClass	1000 1001	VehicleClass Trailer present, CCC=000, class 3 (HV with 4 axles).
42	AttributeId INTEGER(0..127,...)	0001 0011	AttributeId = 19 ₁₀ = VehicleAxles
43	Attribute Value CONTAINER {	0011 0011	Container choice = 51 ₁₀
44	VehicleAxles	0000 0000	VehicleAxles value. Example: vehicle first axle height = not
45		0000 0011	specified, tyre type = not specified, 3 axles (without trailer).
46	AttributeId INTEGER(0..127,...)	0001 0110	AttributeId = 22 ₁₀ = VehicleSpecificCharacteristics
47	Attribute Value CONTAINER {	0011 0110	Container choice = 54 ₁₀
48	EnvironmentalCharacteristics(Euro type, Cop type)	0000	Example : no entry = 0
49	EngineCharacteristics	0000 0000	Example : no entry = 0
50	DescriptiveCharacteristics	0000 0000	Example : no entry = 0
51	FutureCharacteristics	0000 0000	Example : no entry = 0
52	AttributeId INTEGER(0..127,...)	0001 1000	AttributeId = 24 ₁₀ = EquipmentOBUId
53	Attribute Value CONTAINER {	0011 1000	Container choice = 56 ₁₀
54	Equipment OBU Id	0000 0100	OCTET STRING. Example : Kapsch's format (length indicator = 4, manufacturing year = 03, manufacturing month = 05, Serial number = 640)
55		0000 0011	
56		0000 0101	
57		0000 0010	
58		1000 0000	
59	AttributeId INTEGER(0..127,...)	0001 1010	AttributeId = 26 ₁₀ = EquipmentStatus
60	Attribute Value CONTAINER {	0011 1010	Container choice = 58 ₁₀
61	EquipmentStatus	0000 0000	EquipmentStatus (transaction counter). Example : Local use =
62		0011 1010	
63	AttributeId INTEGER(0..127,...)	0010 0001	AttributeId = 33 ₁₀ = ReceiptData1
64	Attribute Value CONTAINER {	0100 0001	Container choice = 65 ₁₀
65	ReceiptData1	0001 1010	ReceiptData1.SessionTime. Example : 2003-03-01, 21:12:10
66		0110 0001	
67		1010 1001	

Octet #	Attribute / Field	Bits in Octet b ₇ b ₀	Description
68		1000 0101	
69		xxxx xxxx	ReceiptData1.SessionServiceProvider
70		xxxx xxxx	
71		xxxx xxxx	
72		xxxx xxxx	ReceiptData1.StationLocation
73		xxxx xxxx	
74		1000 0111	ReceiptData1.SessionLocation. Example: Ascending kilometrage and passage station type.
75		xxxx xxxx	ReceiptData1.SessionType
76		xxxx xxxx	ReceiptData1.SessionResult
77		xxxx xxxx	ReceiptData1.SessionTariffClass
78		xxxx xxxx	ReceiptData1.ClaimedClass
79		xxxx xxxx	ReceiptData1.SessionFee
80		xxxx xxxx	
81		xxxx xxxx	
82		xxxx xxxx	
83		1100 0000	ReceiptData1.SessionServiceProvider (from EFC-ContextMark). Example: Austria, 1 ₁₀ , ASFINAG
84		0100 0000	
85		0000 0001	
86		0000 0011	ReceiptData1.SessionType of contract (from EFC-ContextMark). Example = 3 ₁₀
87		0000 0000	
89		0000 0000	ReceiptData1.SessionContext version (from EFC-ContextMark).
90		xxxx xxxx	ReceiptData1.SessionAuthenticator
91		xxxx xxxx	
92		xxxx xxxx	
93		Xxxx xxxx	
94	FCS	Xxxx xxxx	Frame check sequence
95		xxxx xxxx	
96	FLAG	0111 1110	End Flag

2.2.3 Receipt

Set receipt	Attribute / Field	Bits in Octet b ₇ b ₀	Description	
1	FLAG	0111 1110	Start Flag	
2	Private LID	xxxx xxx0	Link address of a specific OBE	
3		xxxx xxx0		
4		xxxx xxx0		
5		xxxx xxx1		
6	MAC control field	1010 s000	The frame contains a command LPDU	
7	LLC control field	n111 0111	Polled ACn command, n bit	
8	Fragmentation header	1xxx x001	No fragmentation. First service of chain.	
9	GET_STAMPED.requestSEQUENCE {	0000 0101	ACTION.request (GET Stamped, AccessCredential not present, ActionParameter present, IID not present and Reply expected) or alternatively: ACTION.request (GET Stamped, AccessCredentials, ActionParameter present, IID not present and Reply expected)	
		0000 1101		
10	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBE. Example : 5 ₁₀	
11	ActionType INTEGER(0..127,...)	0000 0000	No extension, GET_STAMPED.request = 0	
A	AccessCredential OCTET STRING { AC_CR	0000 0100	No extension, octet string length = 4 ₁₀ Access credential calculated by RSE using RndOBE and the Access Credential Key AC_CRKey. Example: AC_CR(0) = 04 94 F8 97'H	
B		0000 0100		
C		1001 0100		
D		1111 1000		
E		1001 0111		
12	ActionParameter CONTAINER {	0001 0001	No extension, Choice 17 ₁₀ = GetStampedRq	
13	AttributeIdList SEQUENCE (0..127,...) OF { INTEGER (0..127,...) AttributeId {	0000 0001	No extension, number of attribute IDs = 1	
14	PaymentMeans }	0010 0000	AttributeId = 32 ₁₀ = PaymentMeans	
15	Nonce OCTET STRING {	0000 0100	No extension, octet string length = 4 ₁₀	
16	RndRSE	rrrr rrrr	Random number from RSE, containing SessionTime, needed to calculate the Authenticator	
17		rrrr rrrr		
18		rrrr rrrr		
19		rrrr rrrr		
20	KeyRef }	0110 1111	KeyRef_CI (e.g. =111 ₁₀)	
21	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as before (concatenation).	
22	GET.request SEQUENCE {	0110	GET.request	
		OPTION indicator	0	AccessCredential not present or alternatively: AccessCredential present
		OPTION indicator	1	
		OPTION indicator	0	IID not present
		OPTION indicator	1	AttributeIdList present
	Fill BIT STRING(SIZE(1))	0	Set to 0	
23	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBE. Example : 5 ₁₀	
A	AccessCredential OCTET STRING { AC_CR	0000 0100	No extension, octet string length = 4 ₁₀ Access credential calculated by RSE using RndOBE and the Access Credential Key AC_CRKey. Example: AC_CR(0) = 04 94 F8 97'H	
B		0000 0100		
C		1001 0100		
D		1111 1000		
E		1001 0111		
24	AttributeIdList SEQUENCE (0..127,...) OF { { INTEGER (0..127,...) AttributeId {	0000 0001	No extension, number of attribute Ids = 1 ₁₀	
25	VehicleLicencePlateNumber }	0001 0000	AttributeId = 16 ₁₀ = VehicleLicencePlateNr	
26	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as before (concatenation).	
27	SET.request SEQUENCE {	0100 0001	SET.request (No AccessCredential, no IID, fill, reply expected) or alternatively: SET.request (AccessCredential, no IID, fill, reply expected)	
		0100 1001		
28	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBE. Example : 5 ₁₀	
A	AccessCredential OCTET STRING { AC_CR	0000 0100	No extension, octet string length = 4 ₁₀ Access credential calculated by RSE using RndOBE and the Access Credential Key AC_CRKey. Example: AC_CR(0) = 04 94 F8 97'H	
B		0000 0100		
C		1001 0100		
D		1111 1000		
E		1001 0111		
29	AttributeList SEQUENCE ((0..127,...) OF { Attributes SEQUENCE {	0000 0011	No extension, number of attributes in list = 3 ₁₀	
30	AttributeId INTEGER(0..127,...)	0001 1010	AttributeId = 26 ₁₀ = EquipmentStatus	
31	Attribute Value CONTAINER {	0011 1010	Container choice = 58 ₁₀	
32	EquipmentStatus	0000 0000	EquipmentStatus (transaction counter). Example : Local use =	
33		0011 1011	Not specified = 0 ; transaction counter = 59 (58 + 1).	
34	AttributeId INTEGER(0..127,...)	0010 0001	AttributeId = 33 ₁₀ = ReceiptData1	
35	Attribute Value CONTAINER {	0100 0001	Container choice = 65 ₁₀	
36	ReceiptData1	xxxx xxxx	ReceiptData1.SessionTime	
37		xxxx xxxx		

Set receipt	Attribute / Field	Bits in Octet b7 b0	Description
38		xxxx xxxx	
39		xxxx xxxx	
40		xxxx xxxx	ReceiptData1.SessionServiceProvider
41		xxxx xxxx	
42		xxxx xxxx	
43		xxxx xxxx	ReceiptData1.StationLocation
44		xxxx xxxx	
45		xxxx xxxx	ReceiptData1.SessionLocation.
46		xxxx xxxx	ReceiptData1.SessionType
47		xxxx xxxx	ReceiptData1.SessionResult
48		xxxx xxxx	ReceiptData1.SessionTariffClass
49		xxxx xxxx	ReceiptData1.ClaimedClass
50		xxxx xxxx	ReceiptData1.SessionFee
51		xxxx xxxx	
52		xxxx xxxx	
53		xxxx xxxx	
54		xxxx xxxx	ReceiptData1.SessionContractProvider
55		xxxx xxxx	
56		xxxx xxxx	
57		xxxx xxxx	ReceiptData1.SessionTypeOfContract
58		xxxx xxxx	
59		xxxx xxxx	ReceiptData1.SessionContextVersion
60		xxxx xxxx	ReceiptData1.Authenticator
61		xxxx xxxx	
62		xxxx xxxx	
63		xxxx xxxx	
64	AttributeId INTEGER(0..127,...)	0010 0010	AttributeId = 34 ₁₀ = ReceiptData2
65	Attribute Value CONTAINER {	0100 0010	Container choice = 66 ₁₀
66	ReceiptData2	xxxx xxxx	ReceiptData2. Same format as ReceiptData1
....		
93	}	xxxx xxxx	
94	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as before (concatenation).
95	SET_MMI.request SEQUENCE {	0000 0101	ACTION.request (no AccCred, ActionPar, no IID, confirmed, reply)
96	EID INTEGER(0..127,...)	0000 0000	No extension, EID = 0 (system element)
97	ActionType INTEGER(0..127,...)	0000 1010	No extension, SET_MMI.request = 10 ₁₀
98	ActionParameter CONTAINER	xxxx xxxx	No extension, Use Type 69 (preferred) or Type 0 depending on setting in issuer list at RSE
99	SetMMI INTEGER }	0000 0000	Example : ok (0 ₁₀)
100	FCS	xxxx xxxx	Frame check sequence
101		xxxx xxxx	
102	FLAG	0111 1110	End Flag

2.2.3.1 Set receipt response

Octet #	Attribute / Field	Bits in Octet b7 b0	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	1101 0000	The frame contains a response LPDU
7	LLC control field	n111 0111	Acn command n bit
8	LLC status field	0000 0000	Response available and command accepted
9	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as in the corresponding request. .
10	GET_STAMPED.response SEQUENCE {	0001 0100	ACTION.response (Get Stamped rs)
11	EID INTEGER (0..127,...)	0000 0101	Uniquely associated with a context mark in the OBE. Example : 5 ₁₀
12	ResponseParameter CONTAINER {	0001 0010	No extension. Choice 18 ₁₀ = GetStampedRs
13	AttributeList SEQUENCE (0..127,...) OF {	0000 0001	No extension, number of attributes: 1
14	Attributes SEQUENCE { AttributeId	0010 0000	PaymentMeans = 32 ₁₀

Octet #	Attribute / Field	Bits in Octet b ₇ b ₀	Description	
15	AttributeValue CONTAINER {	0100 0000	Container Choice: 64 ₁₀ = PaymentMeans	
16	PersonalAccountNumber	xxxx xxxx	PersonalAccountNumber	
17		xxxx xxxx		
18		xxxx xxxx		
19		xxxx xxxx		
20		xxxx xxxx		
21		xxxx xxxx		
22		xxxx xxxx		
23		xxxx xxxx		
24		xxxx xxxx		
25		xxxx xxxx		
26	PaymentMeansExpiryDate	0001 1110	DateCompact. Example : 2005-03-01	
27		0110 0001		
28	PaymentMeansUsageControl	0000 0000	Example : Not specified 0 ₁₀	
29	}	0000 0000		
30	Authenticator OCTET STRING {	0000 0100	No extension, octet string size = 4 ₁₀	
31	Authenticator	xxxx xxxx	Authenticator over AttributeList (containing PaymentMeans) and RndRSE (containing SessionTime) calculated using AuKey	
32		xxxx xxxx		
33		xxxx xxxx		
34	}	xxxx xxxx		
35	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as in the corresponding request.	
36	GET.response SEQUENCE	0111 0100	GET.response	
37	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBE. Example : 5 ₁₀	
38	AttributeList SEQUENCE (0..127,...) OF {	0000 0001	No extension, 1 attributes in list.	
39	AttributeId INTEGER(0..127,...)	0001 0000	AttributeId = 16 ₁₀ = VehicleLicencePlateNo	
40	Attribute Value CONTAINER {	0010 1111	Container choice = 47 ₁₀	
41	Vehlpn {SEQUENCE countryCode,	1010 0100	VehicleLicencePlateNumber. Example : Country: SE, alphabet indicator no 1 length indicator = 10 chars OCD560, padded with 4 'NUL'-characters	
42		00		
43	AlphabetIndicator,	00 0000		
44	LicencePlateNumber	0000 1010		
45		0100 1111		
46		0100 0011		
47		0100 0100		
48		0011 0101		
49		0011 0110		
50		0011 0000		
51		0000 0000		
52		0000 0000		
53	}	0000 0000		
54	Fragmentation header	1xxx x001		No fragmentation. Same PDU # as in the corresponding request.
55	SET.response SEQUENCE {	0101		SET.response
	OPTION indicator	0		IID not present
	OPTION indicator	0	ResponseStatus not present	
	Fill BIT STRING (SIZE(2))	00	Set to 0	
56	EID INTEGER (0..127,...) }	0000 0101	Uniquely associated with a context mark in the OBE. Example : 5 ₁₀	
57	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as in the corresponding request.	
58	ACTION.response SEQUENCE {	0001	SET_MML.response	
	OPTION indicator	0	IID not present	
	OPTION indicator	0	ResponseParameter not present	
	OPTION indicator	0	ResponseStatus not present	
	Fill BIT STRING (SIZE(1))	0	Set to 0	
59	EID INTEGER (0..127,...) }	0000 0000	No extension, System Element EID = 0	
60	FCS	xxxx xxxx	Frame check sequence	
61		xxxx xxxx		
62	FLAG	0111 1110	End Flag	

2.2.4 Tracking and closing

2.2.4.1 Tracking request (Echo.request)

Octet #	Attribute / Field	Bits in Octet b ₇ b ₀	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	1010 s000	The frame contains a command LPDU
7	LLC control field	n111 0111	Polled ACn command n bit
8	Fragmentation header	1xxx x001	No fragmentation.
9		0000	ACTION.request
	ECHO.request SEQUENCE {		
	OPTION indicator	0	No Access Credentials
	OPTION indicator	1	ActionParameter present
	OPTION indicator	0	IID not present
	Mode BOOLEAN	1	Reply expected
10	EID INTEGER (0..127,...)	0000 0000	No extension, EID = 0
11	ActionType INTEGER (0..127,...)	0000 1111	No extension, ECHO.request = 15
12	ActionParameter CONTAINER	0000 0010	No extension, Choice 2 = Octet string
13	}	0000 0000	No extension. String length = 0 octets
14	FCS	xxxx xxxx	Frame check sequence
15		xxxx xxxx	
16		0111 1110	
16	FLAG	0111 1110	End Flag

2.2.4.2 Tracking response (Echo.response)

Octet #	Attribute / Field	Bits in Octet b ₇ b ₀	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	1101 0000	The frame contains a response LPDU
7	LLC control field	n111 0111	ACn command n bit
8	LLC status field	0000 0000	Response available and command accepted
9	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as in the corresponding request.
10		0001	ACTION.response
	ECHO.response SEQUENCE {		
	OPTION indicator	0	No IID
	OPTION indicator	1	ResponseParameter present
	OPTION indicator	0	ResponseStatus not present
	FILL BIT STRING (SIZE(1))	0	Set to 0.
11	EID INTEGER (0..127,...)	0000 0000	No extension, EID = 0
12	ResponseParameter CONTAINER	0000 0010	No extension, Choice 2 = Octet string
13	}	0000 0000	No extension. String length = 0 octets
14	FCS	xxxx xxxx	Frame check sequence
15		xxxx xxxx	
16	FLAG	0111 1110	End Flag

2.2.4.3 Closing

Octet #	Attribute / Field	Bits in Octet b ₇ b ₀	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	1000 0000	The frame contains a command LPDU
7	LLC control field	0000 0011	UI command
8	Fragmentation header	1xxx x001	No fragmentation.
9	RELEASE.request SEQUENCE {	0010	EVENT_REPORT.request
	OPTION indicator	0	AccessCredential not present
	OPTION indicator	0	EventParameter not present
	OPTION indicator	0	IID not present
	Mode BOOLEAN	0	No reply expected
10	EID INTEGER (0..127,...)	0000 0000	No extension, EID = 0 (system element)
11	EventType INTEGER (0..127,...) }	0000 0000	No extension, RELEASE = 0.
12	FCS	xxxx xxxx	Frame check sequence
13		xxxx xxxx	
14	FLAG	0111 1110	End Flag

3 Enforcement transaction

The EasyGo+/ EETS enforcement transaction will be used by dedicated enforcement equipment where no tolling functionality is needed, e.g. at mobile enforcement equipment.

In contrast to the combined “tolling and enforcement transaction”, only a subset of attributes and the receipt data are read.

3.1 Enforcement transaction - bit-level specification

3.1.1 Initialisation

The Initialisation phase is the same as for the EasyGo+/EETS Tolling and Enforcement Transaction, see chapter 2.2.1

3.1.2 Presentation 1

Similar to the Presentation Phase for the EasyGo+ /EETS Tolling and Enforcement Transaction, but only a subset of data is read-out.

3.1.2.1 Presentation 1 request

Octet #	Attribute / Field	Bits in Octet b ₇ b ₀	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	1010 s000	The frame contains a command LPDU
7	LLC control field	n111 0111	Polled AC _n command, n bit
8	Fragmentation header	1xxx x001	No fragmentation. First service of chain.
9	GET.request SEQUENCE {	0110	GET.request
	OPTION indicator	0	AccessCredential not present or alternatively:
		1	AccessCredential present
	OPTION indicator	0	IID not present
	OPTION indicator	1	AttributeIdList present
	Fill BIT STRING(SIZE(1))	0	Set to 0
10	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBE. Example : 5 ₁₀
A	AccessCredential OCTET STRING { AC_CR }	0000 0100	No extension, octet string length = 4 ₁₀
B		0000 0100	Access credential calculated by RSE using RndOBE and the
C		1001 0100	Access Credential Key AC_CRKey. Example: AC_CR(0) =
D		1111 1000	04 94 F8 97'H
E		1001 0111	
11	AttributeIdList SEQUENCE (0..127,...) OF		
	{ INTEGER (0..127,...) AttributeId {	0000 0110	No extension, number of attribute Ids = 6 ₁₀
	VehicleLicencePlateNumber	0001 0000	AttributeId = 16 ₁₀ = VehicleLicencePlateNr
	VehicleClass	0001 0001	AttributeId = 17 ₁₀ = VehicleClass
	VehicleAxles	0001 0011	AttributeId = 19 ₁₀ = VehicleAxles
	VehicleSpecificCharacteristics	0001 0110	AttributeId = 22 ₁₀ = VehicleSpecificCharacteristics
	EquipmentOBUId	0001 1000	AttributeId = 24 ₁₀ = EquipmentOBUId
	EquipmentStatus } } }	0001 1010	AttributeId = 26 ₁₀ = EquipmentStatus
18	FCS	xxxx xxxx	Frame check sequence
19		xxxx xxxx	
20	FLAG	0111 1110	End Flag

