

Second edition
2011-10-15

AMENDMENT 1
2015-07-01

**Electronic fee collection — Application
interface definition for dedicated
short-range communication**

AMENDMENT 1

*Perception du télépéage — Définition de l'interface d'application
relative aux communications dédiées à courte portée*

AMENDEMENT 1

STANDARDSISO.COM : Click to view the full PDF of ISO 14906:2011/Amd 1:2015



Reference number
ISO 14906:2011/Amd.1:2015(E)

© ISO 2015

STANDARDSISO.COM : Click to view the full PDF of ISO 14906:2011/Amd 1:2015



COPYRIGHT PROTECTED DOCUMENT

© ISO 2015, Published in Switzerland

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

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Foreword

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

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

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

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

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

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

STANDARDSISO.COM : Click to view the full PDF of ISO 14906:2011/Amd 1:2015

Electronic fee collection — Application interface definition for dedicated short-range communication

AMENDMENT 1

1 Amendment to Clause 3.2, action

Page 2, Clause 3.2, delete the clause.

2 Amendment to Clause 3.11, element

Page 3, Clause 3.11, replace:

element

with

Element

3 Amendment to Clause 6.2.3, ContextVersion

Page 14, Clause 6.2.3, replace:

```
EFC-ContextMark:: = SEQUENCE{
    ContractProvider      Provider,
    TypeOfContract        OCTET STRING (SIZE(2)),
    ContextVersion        INTEGER(0..127,..)
}
```

with

```
EFC-ContextMark:: = SEQUENCE{
    ContractProvider      Provider,
    TypeOfContract        OCTET STRING (SIZE(2)),
    ContextVersion        INTEGER(0..127,..)
}
```

The extensibility of the ContextVersion should not be used. ContextVersion is coded as a single octet.

4 Amendment to Clause 8.1, General

Page 35, Clause 8.1, replace:

Table 36 — EFC Attributes

AttributeID	Attribute	Length in Octet	Data Group
0	EFC-ContextMark	6	Contract
1	ContractSerialNumber	4	
2	ContractValidity	6	

Table 36 (continued)

AttributeID	Attribute	Length in Octet	Data Group
35	ValidityOfContract	4	Receipt
3	ContractVehicle	Variable	
4	ContractAuthenticator	Variable	
5	ReceiptServicePart	13	
6	SessionClass	2	
7	ReceiptServiceSerialNumber	3	
36	ReceiptFinancialPart	23	
9	ReceiptContract	9	
10	ReceiptOBUId	Variable	
11	ReceiptICC-Id	Variable	
12	ReceiptText	Variable	
13	ReceiptAuthenticator	Variable	
14	ReceiptDistance	3	
33	ReceiptData1	28	
34	ReceiptData2	28	
15	VehicleIdentificationNumber	Variable	Vehicle
16	VehicleLicencePlateNumber	Variable	
17	VehicleClass	1	
18	VehicleDimensions	3	
19	VehicleAxles	2	
20	VehicleWeightLimits	6	
21	VehicleWeightLaden	2	
22	VehicleSpecificCharacteristics	4	
23	VehicleAuthenticator	Variable	
37	AxleWeightLimits	10	
38	PassengerCapacity	2	
39	Engine	4	Equipment
40	SoundLevel	2	
41	ExhaustEmissionValues	8	
42	DieselEmissionValues	4	
43	CO2EmissionValue	2	
44	VehicleTotalDistance	4	
45	TrailerLicencePlateNumber	Variable	
46	TrailerCharacteristics	5	
24	EquipmentOBUId	Variable	
25	EquipmentICC-Id	Variable	
26	EquipmentStatus	2	Driver
27	DriverCharacteristics	2	
47	ActualNumberOfPassengers	1	

STANDARDSISO.COM Click to view the full PDF of ISO 14906:2011/Amd 1:2015

Table 36 (continued)

AttributeID	Attribute	Length in Octet	Data Group
32	PaymentMeans	14	Payment
29	PaymentMeansBalance	3	
30	PaymentMeansUnit	2	
31	PaymentSecurityData	Variable	
48-53	ReservedForCCC		
54	ReservedForLAC		
55-86	ReservedForFutureCENuse		
87-127	ReservedForPrivateUse		

with

Table 36 — EFC Attributes

AttributeID	Attribute	Length in Octet	Data Group
0	EFC-ContextMark	6	Contract
1	ContractSerialNumber	4	
2	ContractValidity	6	
35	ValidityOfContract	4	
3	ContractVehicle	Variable	
4	ContractAuthenticator	Variable	
5	ReceiptServicePart	13	Receipt
6	SessionClass	2	
7	ReceiptServiceSerialNumber	3	
36	ReceiptFinancialPart	23	
9	ReceiptContract	9	
10	ReceiptOBUId	Variable	
11	ReceiptICC-Id	Variable	
12	ReceiptText	Variable	
13	ReceiptAuthenticator	Variable	
14	ReceiptDistance	3	
33	ReceiptData1	28	
34	ReceiptData2	28	
15	VehicleIdentificationNumber	Variable	
16	VehicleLicencePlateNumber	Variable	
17	VehicleClass	1	
18	VehicleDimensions	3	
19	VehicleAxles	2	
20	VehicleWeightLimits	6	
21	VehicleWeightLaden	2	
22	VehicleSpecificCharacteristics	4	
23	VehicleAuthenticator	Variable	
37	AxleWeightLimits	10	

Table 36 (continued)

AttributeID	Attribute	Length in Octet	Data Group	
38	PassengerCapacity	2		
39	Engine	4		
40	SoundLevel	2		
41	ExhaustEmissionValues	8		
42	DieselEmissionValues	4		
43	CO2EmissionValue	2		
44	VehicleTotalDistance	4		
45	TrailerLicencePlateNumber	Variable		
46	TrailerCharacteristics	5		
55	VehicleCurrentMaxTrainWeight	2		
24	EquipmentOBUId	Variable		Equipment
25	EquipmentICC-Id	Variable		
26	EquipmentStatus	2		
27	DriverCharacteristics	2	Driver	
47	ActualNumberOfPassengers	1		
32	PaymentMeans	14	Payment	
29	PaymentMeansBalance	3		
30	PaymentMeansUnit	2		
31	PaymentSecurityData	Variable		
48-53	ReservedForCCC			
54	ReservedForLAC			
56-86	ReservedForFutureCENuse			
87-127	ReservedForPrivateUse			

5 Amendment to Table 39 — Data group Vehicle

Page 44, Table 39, replace Table 39 with the following:

Table 39 — Data group Vehicle

EFC Attribute	Data element	Definition	Type	Length in octet	Value Range	Informative remarks
VehicleLicence PlateNumber	VehicleLicence PlateNumber	Claimed licence plate of the vehicle	LPN	Variable		
VehicleIdentificationNumber	VehicleIdentificationNumber	Identification number of vehicle according ISO 3779	CS5	Variable		Imported from ISO 14816
VehicleClass	VehicleClass	Service provider specific information pertaining to the vehicle.	INT1	1		
VehicleDimensions	VehicleLengthOverall	Nominal maximum overall length of the vehicle according to ISO 612, in dm, rounded to the next dm.	INT1	1		
	VehicleHeightOverall	Nominal overall unladen height, according to ISO 612, in dm, rounded to the next dm.	INT1	1		
	VehicleWidthOverall	Nominal overall width, according to ISO 612, in dm, rounded to the next dm	INT1	1		
VehicleAxles	VehicleFirstAxleHeight	Bonnet height, measured over the front axle, in dm, rounded to the next dm.	INT1	1		
	VehicleAxlesNumber	Tyre type and number of axles, including drop axles.	VehicleAxles	1		2 bits for dual tyre. 6 bits are used for the definition of number of axles: 3 bits to encode the number of all axles of the tractor and 3 to encode the number of all axles of the trailer.
VehicleWeight Limits	VehicleMaxLadenWeight	Maximum permissible total weight including payload, according to ISO 1176. 10 kg units, rounded down to the next 10kg step.				Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 (LSB)
						Nr of axles of Trailer
						0 to 7
						0 to 7
						See Note 2.
			INT2	2		

Table 39 (continued)

EFC Attribute	Data element	Definition	Type	Length in octet	Value Range	Informative remarks
	VehicleTrainMaximumWeight	Maximum permissible weight of the complete vehicle train, as defined in ISO 1176. 10 kg units, rounded down to the next 10kg step.	INT2	2		ISO 1176 Code ISO-M18 maximum design mass of vehicle combination
	VehicleWeightUnladen	Nominal unladen weight, according to ISO 1176 in 10 kg units, rounded down to the next 10 kg step.	INT2	2		
VehicleWeightLaden	VehicleWeightLaden	Actual weight of vehicle including load in 10kg units, rounded down to the next 10 kg step.	INT2	2		
VehicleCurrentMaxTrainWeight	VehicleCurrentMaxTrainWeight	Maximum permissible weight of the complete vehicle train that is currently in operation, which shall be as defined in ISO 1176-10 kg units, rounded down to the next 10kg step.	INT2	2		This weight may be lower than VehicleTrainMaximumWeight as it represents the current maximum train weight and not the maximum design mass.
AxleWeightLimits	MaxLadenWeightOnAxle1	Technically permissible maximum laden weight on axle 1 of the vehicle, 10 kg units, rounded down to the next 10 kg step.	INT2	2		
	MaxLadenWeightOnAxle2	Technically permissible maximum laden weight on axle 2 of the vehicle, 10 kg units, rounded down to the next 10 kg step.	INT2	2		
	MaxLadenWeightOnAxle3	Technically permissible maximum laden weight on axle 3 of the vehicle, 10 kg units, rounded down to the next 10 kg step.	INT2	2		
	MaxLadenWeightOnAxle4	Technically permissible maximum laden weight on axle 4 of the vehicle, 10 kg units, rounded down to the next 10 kg step.	INT2	2		
	MaxLadenWeightOnAxle5	Technically permissible maximum laden weight on axle 5 of the vehicle, 10 kg units, rounded down to the next 10 kg step.	INT2	2		
PassengerCapacity	NumberOfSeats	Number of seats of the vehicle, including the driver's seat.	INT1	1	0...255	

Table 39 (continued)

EFC Attribute	Data element	Definition	Type	Length in octet	Value Range	Informative remarks
	Number of Standing Places	Number of standing places of the vehicle	INT1	1	0...255	
Vehicle Specific Characteristics	Vehicle Specific Characteristics	Further vehicle characteristics. Each enumerated value has a specific meaning assigned. The meaning of some values are defined in this standard, others are reserved for future needs.	Vehicle Specific Characteristics	4		Assignment of meaning to the unassigned enumerated values is subject to registration according to the registration procedure specified in EN 12834 / ISO 15628.
Engine	Engine Capacity	Capacity of the vehicle's engine in cm ³	INT2	2		
	Engine Power	Maximum net power of the vehicle's engine, in KW	INT2	2		
Sound Level	Sound Stationary	Stationary Sound of the vehicle, according to vehicle registration documents in dB(A)	INT1	1	0...255	
	Sound Drive By	Sound of the vehicle when driving, according to vehicle registration documents in dB(A)	INT1	1	0...255	
Exhaust Emission Values	Emission CO	Exhaust emission of CO, according to vehicle registration documents, in 10 ⁻³ g/km or g/kWh.	INTE-GER(0...32766)			If the emissions are measured directly on the engine test bed the value is declared in g/kWh
	Emission HC	Exhaust emission of HC, according to vehicle registration documents, in 10 ⁻³ g/km or g/kWh.	INT 2	2	0...65535	If the emissions are measured directly on the engine test bed the value is declared in g/kWh
	Emission NOX	Exhaust emission of NOX, according to vehicle registration documents, in 10 ⁻³ g/km or g/kWh.	INT 2	2	0...65535	If the emissions are measured directly on the engine test bed the value is declared in g/kWh
	Emission HCNOX	Exhaust emission of HCNOX, according to vehicle registration documents, in 10 ⁻³ g/km or g/kWh.	INT 2	2	0...65535	If the emissions are measured directly on the engine test bed the value is declared in g/kWh
Diesel Emission Values	Particulate	Particulates for diesel, according to vehicle registration documents, in 10 ⁻³ g/km or g/kWh.	INTE-GER(0...32766)	2		If the emissions are measured directly on the engine test bed the value is declared in g/kWh
	Absorption Coeff	Corrected absorption coefficient for diesel, according to vehicle registration documents, in 10 ⁻³ m ⁻¹ .	INT 2	2	0...65535	

Table 39 (continued)

<i>EFC Attribute</i>	<i>Data element</i>	<i>Definition</i>	<i>Type</i>	<i>Length in octet</i>	<i>Value Range</i>	<i>Informative remarks</i>
CO2EmissionValue	CO2EmissionValue	Vehicle's CO ₂ emission value according to vehicle registration documents, in g/km.	INT2	2	0...65535	
VehicleTotalDistance	VehicleTotalDistance	Total distance as measured by the vehicle, in 10 m resolution, continuously incremented.	INT4	4	0...4294967294	The initial value of this attribute may be either the value zero or the vehicle's kilometer reading at the moment of personalisation of the OBU
TrailerLicencePlateNumber	TrailerLicencePlateNumber	Claimed licence plate of the trailer	LPN	Variable		
TrailerCharacteristics	TrailerDetails	Indication provided on trailer presence, type and number of axles.	TrailerDetails	1		5 bits are used for the trailer presence and type. 3 bits are used for the number of axles. If only one trailer is present, the presence and the number of axles of this single trailer is available in VehicleAxles and may not be included in this attribute.
	TrailerMaxLadenWeight	Maximum permissible total weight of the trailer including payload, according to ISO 1176-10 kg units, rounded down to the next 10 kg step.	INT2	2		
	TrailerWeightUnladen	Nominal unladen weight of the trailer, according to ISO 1176 in 10 kg units, rounded down to the next 10 kg step.	INT2	2		
VehicleAuthenticator	VehicleAuthenticator	Authenticator calculated by the entity entering the data elements at time of entry or modification.	OCTET STRING	Variable		

6 Amendment to Annex A

Replace the existing ASN.1 code with:

```

EfcDsrcApplication {iso(1) standard(0) 14906 application(0) version5(5)}
DEFINITIONS AUTOMATIC TAGS ::= BEGIN
EXPORTS ALL;

IMPORTS

CountryCode, CS5, IssuerIdentifier
FROM AVIAEINumberingAndDataStructures {iso(1) standard(0) 14816}
-- defined in ISO 14816:-

AttributeIdList, Attributes{}, AttributeList{}
FROM EfcDsrcGeneric {iso(1) standard(0) 14906 generic(1) version5(5)};
-- NOTE: The following are the definitions of the action and response
-- parameters

ActualNumberOfPassengers ::= Int1

AxleWeightLimits ::= SEQUENCE {
    maxLadenweightOnAxle1 Int2,
    maxLadenweightOnAxle2 Int2,
    maxLadenweightOnAxle3 Int2,
    maxLadenweightOnAxle4 Int2,
    maxLadenweightOnAxle5 Int2
}

AddRq ::= SEQUENCE {
    attributeId INTEGER (0..127, ..),
    value INTEGER
}

ChannelId ::= INTEGER {
    obu (0),
    sam1 (1), -- secure application module
    sam2 (2),
    icc (3), -- integrated circuit(s) card
    display (4),
    buzzer (5),
    printer (6),
    serialInterface (7), -- serial interface: e.g. RS232 and RS485
    parallelInterface (8),
    gPS (9),
    tachograph (10),
    privateUse1 (11), -- free for proprietary use
    privateUse2 (12), -- free for proprietary use
    privateUse3 (13), -- free for proprietary use
    privateUse4 (14), -- free for proprietary use
    privateUse5 (15), -- free for proprietary use
    bluetooth (16)
-- (17-255) are reserved for future CEN use
} (0..255)

ChannelRq ::= SEQUENCE {
    channelId ChannelId,
    apdu OCTET STRING
    -- format according to the interface
    -- of the channelId
}

ChannelRs ::= SEQUENCE {
    channelId ChannelId,
    apdu OCTET STRING
    -- format according to the interface
    -- of the channelId
}

```

```

CopyRq ::= SEQUENCE {
    destinationEID      INTEGER(0..127,...),
    attributeIdList     AttributeIdList
}

CreditRq ::= SEQUENCE {
    refund              PaymentFee,
    nonce              OCTET STRING,
    key                 INTEGER(0..255)
}

CreditRs ::= SEQUENCE {
    creditResult        ResultFin,
    creditAuthenticator OCTET STRING
}

DebitRq ::= SEQUENCE {
    debitPaymentFee     PaymentFee,
    nonce              OCTET STRING,
    keyRef              INTEGER(0..255)
}

DebitRs ::= SEQUENCE {
    debitResult         ResultFin,
    debitAuthenticator OCTET STRING
}

GetInstanceRq ::= SEQUENCE {
    posOfFirstInstance  INTEGER(0..255),
    -- position of first instance to be retrieved
    posOfLastInstance   INTEGER(0..255),
    -- position last instance to be retrieved
    attributeIdList     AttributeIdList,
    -- Ids of attributes to be retrieved
}

GetInstanceRs{Container} ::= SEQUENCE (SIZE (0..127,...)) OF SEQUENCE {
    attributeId          INTEGER(0..127,...),
    -- number of instances retrieved
    attributeValues      Container (WITH COMPONENTS {octetstring PRESENT})
    -- The octetstring shall contain the concatenation of
    -- the unaligned PER encodings of the values of the
    -- instances, with each encoding padded to an integral
    -- of octets as specified for a top-level type in
    -- ISO/IEC 8825-2
}

GetStampedRq ::= SEQUENCE {
    attributeIdList     AttributeIdList,
    nonce              OCTET STRING, -- e.g. a random number
    keyRef              INTEGER(0..255)
}

GetStampedRs{Container} ::= SEQUENCE {
    attributeList       AttributeList{Container},
    authenticator       OCTET STRING
}

SetInstanceRq{Container} ::= SEQUENCE {
    posOfInstance       INTEGER(0..255),
    attribute            Attributes{Container}
}

SetMMIRq ::= INTEGER {
    ok                   (0), -- operation / transaction successfully completed
    nok                  (1), -- operation / transaction not successfully completed
    contactOperator      (2), -- e.g. due to low balance or battery
    noSignalling         (255) -- no signalling
    -- (3-127) are reserved for future CEN use
}

```

```

-- (128-254) are reserved for private use
} (0..255)

SetStampedRq{Container} ::= SEQUENCE {
    attributeList      AttributeList{Container},
    nonce              OCTET STRING,
    keyRef             INTEGER(0..255)
}

SubRq ::= SEQUENCE {
    attributeId        INTEGER(0..127,...),
    value              INTEGER
}

-- NOTE: The following are the definitions of EFC attributes

CO2EmissionValue ::= Int2
ContractSerialNumber ::= Int4
ContractAuthenticator ::= OCTET STRING

ContractValidity ::= SEQUENCE {
    contractRestrictions      OCTET STRING (SIZE(4)),
    contractExpiryDate       DateCompact
} -- intended to support ENV implemented systems

ContractVehicle ::= LPN

DateCompact ::= SEQUENCE {
    year          INTEGER (1990..2117),
    month         INTEGER (0..12), -- Value zero shall not be used
                -- except with 1990 - see below.
    day          INTEGER (0..31) -- Value zero shall not be used
                -- except with 1990 - see below.
}

-- The value "{year 1990, month 0, day 0}" is a 16-bit all-zero
-- encoding, and is used to represent "no date".

DescriptiveCharacteristics ::= INTEGER {
    noEntry          (0),
    vehicleShape1   (1),
    vehicleShape2   (2),
    vehicleShape3   (3),
    vehicleShape4   (4),
    vehicleShape5   (5),
    vehicleShape6   (6),
    vehicleShape7   (7),
    vehicleShape8   (8),
    vehicleShape9   (9),
    vehicleShape10  (10),
    vehicleShape11  (11),
    vehicleShape12  (12),
    vehicleShape13  (13),
    vehicleShape14  (14),
    vehicleShape15  (15),
    vehicleShape16  (16),
    vehicleShape17  (17),
    vehicleShape18  (18),
    vehicleShape19  (19),
    vehicleShape20  (20),
    vehicleShape21  (21),
    vehicleShape22  (22),
    vehicleShape23  (23),
    vehicleShape24  (24),
    vehicleShape25  (25),
    vehicleShape26  (26),
    vehicleShape27  (27),
    vehicleShape28  (28),
    vehicleShape29  (29),
    vehicleShape30  (30),

```

```

vehicleShape31    (31),
vehicleShape32    (32),
vehicleShape33    (33),
vehicleShape34    (34),
vehicleShape35    (35),
vehicleShape36    (36),
vehicleShape37    (37),
vehicleShape38    (38),
vehicleShape39    (39),
vehicleShape40    (40),
vehicleShape41    (41),
vehicleShape42    (42),
vehicleShape43    (43),
vehicleShape44    (44),
vehicleShape45    (45),
vehicleShape46    (46),
vehicleShape47    (47),
vehicleShape48    (48),
vehicleShape49    (49),
vehicleShape50    (50)
-- (1..50) are reserved for future CEN use
-- (51..255) are reserved for private use
} (0..255)

DieselEmissionValues ::= SEQUENCE {
    particulate SEQUENCE {
        unitType      UnitType,
        value          INTEGER (0..32767)
    },
    absorptionCoeff  Int2    }

DriverCharacteristics ::= SEQUENCE {
    driverClass      Int1,
    tripPurpose      Int1
}

EFC-ContextMark ::= SEQUENCE {
    contractProvider Provider,
    typeOfContract  OCTET STRING (SIZE(2)),
    contextVersion  INTEGER(0..127,...)
}

EnvironmentalCharacteristics ::= SEQUENCE {
    euroValue        EuroValue,
    copValue         CopValue
}

EuroValue ::= ENUMERATED {
    noEntry          (0),
    euro-1           (1),
    euro-2           (2),
    euro-3           (3),
    euro-4           (4),
    euro-5           (5),
    euro-6           (6),
    reservedForUse1 (7),
    reservedForUse2 (8),
    reservedForUse3 (9),
    reservedForUse4 (10),
    reservedForUse5 (11),
    reservedForUse6 (12),
    reservedForUse7 (13),
    reservedForUse8 (14),
    eev              (15)
} -- 4 bits, EURO-Classes as defined in EC directive 88/77/EEC, annex 1
-- and in 91/542/EEC, 96/1/EC, 1999/96/EC, 2001/27/EC, regulation
-- No 595/2009 and for EEV in Section 6.2.1 of Annex I in EC directive
-- 2005/55/EC

CopValue ::= ENUMERATED {
    noEntry          (0),

```

STANDARDSISO.COM Click to view the full PDF of ISO 14906:2011/Amd 1:2015

```

co2class1          (1), - below 101 g/km
co2class2          (2), - 101 to 120 g/km
co2class3          (3), - 121 to 140 g/km
co2class4          (4), - 141 to 160 g/km
co2class5          (5), - 161 to 200 g/km
co2class6          (6), - 201 to 250 g/km
co2class7          (7), - above 250 g/km
reservedforUse (8) -- reserved for future CEN and ISO use
} -- 4 bits, reserved for carbon dioxide pollution values as
  -- defined in EC directive 2003/127/EC'

EngineCharacteristics ::= INTEGER {
noEntry          (0),
noEngine         (1),
petrolUnleaded  (2),
petrolLeaded    (3),
diesel          (4),
LPG             (5),
battery         (6),
solar           (7),
hybrid          (8),
hydrogen        (9)
-- (10-255) are reserved for future CEN use
} (0..255)

Engine ::= SEQUENCE {
  engineCapacity  Int2,
  enginePower     Int2
}

EquipmentOBUID ::= OCTET STRING
EquipmentStatus ::= BIT STRING (SIZE(16))

ExhaustEmissionValues ::= SEQUENCE {
  unitType        UnitType,
  emissionCO      INTEGER (0..32767),
  emissionHC      Int2,
  emissionNOX     Int2,
  emissionHCNOX   Int2
}

FutureCharacteristics ::= INTEGER {
  noEntry          (0),
  airSuspension    (1)
-- (2..255) are reserved for future CEN use
} (0..255)

ICC-Id ::= OCTET STRING
Int1 ::= INTEGER(0..255)
Int2 ::= INTEGER(0..65535)
Int3 ::= INTEGER(0..16777215)
Int4 ::= INTEGER(0..4294967295)

LPN ::= SEQUENCE {
  countryCode      CountryCode,
  alphabetIndicator ENUMERATED {
    latinAlphabetNo1 (1), -- encoded as 00 00 00'B
    latinAlphabetNo2 (2), -- encoded as 00 00 01'B etc
    latinAlphabetNo3 (3),
    latinAlphabetNo4 (4),
    latinCyrillicAlphabet (5),
    latinArabicAlphabet (6),
    latinGreekAlphabet (7),
    latinHebrewAlphabet (8),
    latinAlphabetNo5 (9),
    latinAlphabetNo6 (10),
    twoOctetBMP (11),
    fourOctetCanonical (12),
    reservedForUse1 (13),
    reservedForUse2 (14),
    reservedForUse3 (15),

```

```

reservedForUse4          (16),
reservedForUse5          (17),
reservedForUse6          (18),
reservedForUse7          (19),
reservedForUse8          (20),
reservedForUse9          (21),
reservedForUse10         (22),
reservedForUse11         (23),
reservedForUse12         (24),
reservedForUse13         (25),
reservedForUse14         (26),
reservedForUse15         (27),
reservedForUse16         (28),
reservedForUse17         (29),
reservedForUse18         (30),
reservedForUse19         (31),
reservedForUse20         (32),
reservedForUse21         (33)
} -- 6 bits, latinAlphabetNo1 recommended --,
  -- refer to Annex E for conversion from LatinAlphabetNo 2
  -- and 5 to Latin AlphabetNo1
licencePlateNumber OCTET STRING
}

PassengerCapacity ::= SEQUENCE {
    numberOfSeats      Int1,
    numberOfStandingPlaces Int1
}

PaymentFee ::= SEQUENCE {
    -- The fee (toll, charge or fare) which is requested by the
    -- service provider for the service provided or to be provided.
    paymentFeeAmount Int2,
    -- paymentFeeAmount is the value of the fee being charged for the
    -- service. If no unit (payment fee unit) is specified, then
    -- it is known by default.
    paymentFeeUnit PayUnit
    -- paymentFeeUnit is the unit in which the fee is expressed.
}

PaymentMeans ::= SEQUENCE {
    personalAccountNumber PersonalAccountNumber,
    paymentMeansExpiryDate DateCompact,
    pamentMeansUsageControl OCTET STRING (SIZE(2))
    -- issuer's specified restrictions, on the geographic usage
    -- and services allowed for the applications
}

PaymentMeansBalance ::= SignedValue

SignedValue ::= CHOICE {
    positive      INTEGER (0..8388607),
    negative      INTEGER (-8388608..-1)
}

-- corresponds to a "3 octets Signed Integer", associated with the following
-- examples of line codes:
-- -8'388'608: 80 00 00'H
-- -1: FF FF FF'H
-- 0: 00 00 00'H
-- 1: 00 00 01'H
-- 8'388'607: 7F FF FF'H

PaymentMeansUnit ::= PayUnit
PaymentSecurityData ::= OCTET STRING
PayUnit ::= OCTET STRING (SIZE(2))

-- The unique designation of a Currency as defined in ISO 4217
-- using the ISO numeric binary coded decimal representation.
-- The code can also express a company specific token or a
-- "charging unit code" as used in the freight unit in which

```

```

-- the fee is expressed.
-- Value Assignment:
-- '0xxx'H Currency in main units
-- '1xxx'H Currency in minor units of 10:1 ('dime')
-- '2xxx'H Currency in minor units of 100:1 ('cents')
-- '3xxx'H Currency in minor units of 1000:1
-- '4xxx'H Currency in 'major' units / 10
-- (e.g. 10 Belgian Francs)
-- '5xxx'H Currency in 'major' units / 100
-- (e.g. 100 Italian Lire)
-- '6xxx'H Currency in 'major' units / 1000
-- '7xxx'H Currency in 'major' units / 10000
-- '8xxx'H Currency in 'major' units / 100000
-- where xxx is the BCD representation of "Currency"
-- as defined in ISO 4217
-- '9xxx'H Tokens
-- where xxx is Purse Provider specific coding.
-- 'Axxx'H Charging Unit Codes,
-- denoting quantification of the service provided
-- (e.g. man-hours)

PersonalAccountNumber ::= OCTET STRING (SIZE(10))

-- Personal account number structure - according to ISO/IEC 7812-1
-- Issuer identifier number ("BIN")
-- Major industry identifier (MII, 1 binary coded decimal, BCD)
-- 0: reserved for future use by ISO/IEC 7812-1
-- 1: airline sector
-- 2: extended airline sector
-- 3: travel and tourism sector
-- 4: financial banking sector
-- 5: financial banking sector
-- 6: commerce and banking sector
-- 7: petrol industry sector
-- 8: telecommunication sector
-- 9: reserved for national use
-- Issuer identifier (5 BCD in the second edition of ISO/IEC 7812-1)
-- Account number (max 12 BCD)
-- Control digit (1 BCD)
-- Padding bits, set to 1'B, in order to accomplish a
-- total length of 10 octets.

Provider ::= SEQUENCE {
countryCode CountryCode,
providerIdentifier IssuerIdentifier
}
PurseBalance ::= SEQUENCE {
-- The balance on the (electronic) purse, consisting of
-- the value and the unit in which it is expressed.
purseValue SignedValue,
-- The size of a balance expressed in a currency.
-- This may be positive or negative.
purseUnit PayUnit
}

ReceiptContract ::= SEQUENCE {
sessionContractProvider Provider,
sessionTypeOfContract OCTET STRING(SIZE(2)),
sessionContractSerialNumber Int4
}

ReceiptData1 ::= ReceiptData
ReceiptData2 ::= ReceiptData

ReceiptData ::= SEQUENCE {
sessionTime DateAndTime,
sessionServiceProvider Provider,
locationOfStation Int2,
sessionLocation SessionLocation,
sessionType Int1,
sessionResult ResultOp,
}

```

```

    sessionTariffClass      Intl,
    sessionClaimedClass    Intl,
    sessionFee              PaymentFee,
    sessionContractProvider Provider,
    sessionTypeOfContract  OCTET STRING (SIZE(2)),
    sessionContextVersion  INTEGER (0..127,...),
    receiptDataAuthenticator OCTET STRING(SIZE(4))
}

ReceiptDistance ::=      Int3

ReceiptFinancialPart ::= SEQUENCE {
    personalAccountNumber PersonalAccountNumber,
    sessionPaymentFee      PaymentFee,
    sessionCurrentBalance  PurseBalance,
    receiptFinancialSerialNumber Int4
}

ReceiptICC-Id ::=      ICC-Id
ReceiptOBUID ::=      OCTET STRING

ReceiptServicePart ::= SEQUENCE {
    sessionTime            DateAndTime,
    sessionServiceProvider Provider,
    stationLocation        INTEGER(0..1048575),
    sessionLocation        BIT STRING (SIZE(8)),
    typeOfSession          StationType,
    sessionResultOperational ResultOp,
    sessionResultFinancial ResultFin
}

ReceiptServiceSerialNumber ::= Int3
ReceiptAuthenticator ::= OCTET STRING
ReceiptText ::=      OCTET STRING
ResultFin ::=      OCTET STRING (SIZE(1))

-- A code designating whether a card transaction was completed successfully
-- or not. Value Assignment: Hexadecimal
-- Most significant 4 bits: 0 OK,
-- '0x'H OK
-- Most significant 4 bits > 0 Not OK:
-- '1x'H Not OK, not specified further
-- '2x'H Not OK, Abnormal (First or Previous) Event
-- '3x'H Not OK, Contract not accepted
-- '4x'H Not OK, Account or Purse not accepted
-- 'x0'H not specified further
-- 'x1'H Balance close to zero
-- 'x2'H Balance now negative
-- 'x3'H Balance Overflow
-- 'x4'H Provider not accepted
-- 'x5'H Authentication failure
-- x6'H Vehicle Class incorrect

ResultOp ::=      INTEGER {
    correctTransaction          (0), -- transaction correct
    obeStatusNotAccepted       (1),
    equipmentStatusNotAccepted (2),
    contractNotInWhiteList     (3),
    -- VST contract data not in white list
    contractIdentifierInBlackList (4),
    contractIdentifierNotCorrect (5),
    -- Luhn algorithm verification failure
    expiredContract            (6), -- contract expired
    contractRestrictionsNotFulfilled (7),
    claimedVehicleCharacteristicsNotValid (8),
    vehicleClassAuthenticationFailed (9),
    entryVehicleClassDifferentFromExitVehicleClass (10),
    entryReceiptMissing        (11),
    entryReceiptNotValid       (12),
    entryTollStationNotValid   (13),
    equipmentNotCertified      (14),

```

STANDARDS GROUP - Click to view the full PDF of ISO 14906:2011/Amd 1:2015

```

-- manufacturer or EquipClass not recognised
timeDifference          (15),
-- problem with the time diff of the two latest receipts
accessCredentialsNotAccepted (16),
contractAuthenticatorNotAccepted (17),
receiptAuthenticatorNotAccepted (18),
claimedVehicleCharacteristicsMissing (19),
paymentMeansNotAccepted (20),
paymentAuthenticatorNotAccepted (21),
paymentMeansInBlackList (22),
paymentMeansNotCorrect (23),
-- Luhn algorithm verification failure
expiredPaymentMeans (24),
-- PaymentMeans expired
paymentMeansRestrictionsNotFulfilled (25)
-- (26-255) are reserved for future CEN use
} (0..255)

SessionClass ::= SEQUENCE {
    sessionTariffClass Int1,
    sessionClaimedClass Int1
}

SessionLocation ::= SEQUENCE {
    ascendingKilometrage BOOLEAN, -- travel direction indicator
    laneCodeNumber INTEGER(0..127) -- lane code number
}

StationType ::= ENUMERATED {
    unspecified (0),
    closedEntryWithPayment (1),
    closedEntryWithoutPayment (2),
    closedTransit (3),
    closedExit (4),
    closedCredit (5),
    mixed (6),
    passage (7), -- open exit
    checkpoint (8),
    reload (9),
    reservedForFutureCENUse1 (10),
    reservedForFutureCENUse2 (11),
    reservedForFutureCENUse3 (12),
    reservedForFutureCENUse4 (13),
    privateUse5 (14),
    privateUse6 (15)
}

DateAndTime ::= SEQUENCE {
    timeDate DateCompact,
    timeCompact SEQUENCE { -- expresses time of the day in hours, min,
and sec
        hours INTEGER (0..23), -- number of hours after midnight
        mins INTEGER (0..59), -- number of minutes after the hour
        double-secs INTEGER (0..30) -- number of two-seconds after the
minute
    }
-- Midnight at the start of a day cannot be represented.
-- Midnight at the end of a day is represented by
-- {hours 23, mins 59, double-secs 30}
-- The 16 bit zero value {hours 0, mins 0, double-secs 0}
-- denotes "no time"
}

SoundLevel ::= SEQUENCE {
    soundstationary Int1,
    sounddriveby Int1
}

TrailerCharacteristics ::= SEQUENCE {
    trailerDetails TrailerDetails,
    trailerMaxLadenWeight Int2,

```

```

    trailerWeightUnladen    Int2
    }

TrailerDetails ::= SEQUENCE {
    trailerType             INTEGER{
        notPresent         (0), -- trailer not attached or only one trailer attached,
        trailer            (1), -- VehicleAxlesNumber for more information
                             -- also known as pull-bar trailer
        semitrailer        (2), -- also known as articulate trailer
                             -- (3..31) reserved for future CEN/ISO use
    } (0..31),
    trailerAxles            TrailerAxles
}

TrailerLicencePlateNumber ::= LPN

UnitType ::= ENUMERATED {
    mg-km                   (0),
    mg-kWh                  (1)
}

ValidityOfContract ::= SEQUENCE {
    issuerRestrictions      OCTET STRING (SIZE(2)),
    contractExpiryDate     DateCompact
}

VehicleAuthenticator ::= OCTET STRING

VehicleAxles ::= SEQUENCE {
    vehicleFirstAxleHeight Int1,
    vehicleAxlesNumber     SEQUENCE {
        tyreType           ENUMERATED{
            notSpecified   (0),
            singleTyre     (1), -- single tyre on all axles
            dualTyres      (2), -- dual tyres on at least one axle
            reservedForUse (3) -- reserved for future CEN use
        },
        numberOfAxles      SEQUENCE {
            trailerAxles    TrailerAxles,
            tractorAxles    TractorAxles
        }
    }
}

TrailerAxles ::= INTEGER (0..7) -- number of axles of the trailer when
available
TractorAxles ::= INTEGER (0..7) -- number of axles of the tractor
VehicleClass ::= Int1

VehicleDimensions ::= SEQUENCE {
    vehicleLengthOverall Int1,
    vehicleHeightOverall Int1,
    vehicleWidthOverall Int1
}

VehicleLicencePlateNumber ::= LPN
VehicleIdentificationNumber ::= CS5

VehicleSpecificCharacteristics ::= SEQUENCE {
    environmentalCharacteristics EnvironmentalCharacteristics,
    engineCharacteristics      EngineCharacteristics,
    descriptiveCharacteristics DescriptiveCharacteristics,
    futureCharacteristics      FutureCharacteristics
}

VehicleTotalDistance ::= Int4
VehicleWeightLaden ::= Int2
VehicleCurrentMaxTrainWeight ::= Int2

VehicleWeightLimits ::= SEQUENCE {

```

STANDARD.PDFISO.COM : Click to view the full PDF of ISO 14906:2011/Amd 1:2015

```

vehicleMaxLadenWeight      Int2,
vehicleTrainMaximumWeight  Int2,
vehicleWeightUnladen       Int2
}

```

END

EfcDsrcGeneric {iso(1) standard(0) 14906 generic(1) version5(5)}

DEFINITIONS AUTOMATIC TAGS ::= BEGIN
EXPORTS ALL;

IMPORTS

GetStampedRq, GetStampedRs {}, SetStampedRq {}, GetInstanceRq,
GetInstanceRs {}, SetInstanceRq {}, SetMMIRq, ChannelRq, ChannelRs, CopyRq,
SubRq, AddRq, DebitRq, DebitRs, CreditRq, CreditRs,

EFC-ContextMark, ContractSerialNumber, ContractValidity,
ContractVehicle, ContractAuthenticator, ICC-Id,
ReceiptServicePart, SessionClass, SignedValue, ReceiptServiceSerialNumber,
ReceiptFinancialPart, ReceiptContract, ReceiptOBUId, ReceiptICC-Id,
ReceiptText, ReceiptAuthenticator, ReceiptDistance, LPN, VehicleClass,
VehicleDimensions, VehicleAxles, VehicleWeightLimits,
VehicleWeightLaden, VehicleSpecificCharacteristics,
VehicleAuthenticator, EquipmentOBUId,
EquipmentStatus, DriverCharacteristics,
PaymentMeans, PaymentMeansBalance, PaymentMeansUnit,
PaymentSecurityData, PersonalAccountNumber, ReceiptData1,
ReceiptData2, SessionLocation, ValidityOfContract,
AxleWeightLimits, PassengerCapacity, Engine, SoundLevel,
ExhaustEmissionValues, DieselEmissionValues, CO2EmissionValue,
VehicleTotalDistance, TrailerLicencePlateNumber,
TrailerCharacteristics, ActualNumberOfPassengers
FROM EfcDsrcApplication {iso(1) standard(0) 14906 application(0) version5(5)}

CS5

FROM AVIAEINumberingAndDataStructures {iso(1) standard(0) 14816};
-- defined in ISO 14816:-
-- EXPORTS everything;

```

Action-Request(Container) ::= SEQUENCE{
    mode          BOOLEAN,
    eid           Dsrc-EID,
    actionType    ActionType,
    accessCredentials OCTET STRING (SIZE (0..127,...)) OPTIONAL,
    actionParameter Container OPTIONAL,
    iid           Dsrc-EID OPTIONAL
}

```

```

Action-Response(Container) ::= SEQUENCE{
    fill          BIT STRING (SIZE(1)),
    eid           Dsrc-EID,
    iid           Dsrc-EID OPTIONAL,
    responseParameter Container OPTIONAL,
    ret           ReturnStatus OPTIONAL
}

```

ActionType ::= INTEGER (0..127,...)

```

-- 0: getStamped
-- 1: setStamped
-- 2: getSecure
-- 3: setSecure
-- 4: getInstance
-- 5: setInstance
-- 6: getNonce
-- 7: setNonce
-- 8: transferChannel
-- 9: copy

```

```

-- 10: setMMI
-- 11: subtract
-- 12: add
-- 13: debit
-- 14: credit
-- 15: echo
-- (16..118) Reserved for ISO/CEN use
-- (119-127) Reserved for private use
ApplicationContextMark ::= EfcContainer (WITH COMPONENTS {octetstring PRESENT})
-- The contents of the octetstring shall be an aligned PER
-- encoding of EFC-Contextmark, but this encoding may be followed
-- by non-standardised encodings. Illustrations of the inclusion
-- of non-standardised encodings are shown in Annex B.
ApplicationList ::= SEQUENCE (SIZE (0..127,...)) OF SEQUENCE {
    aid          DsrcApplicationEntityID,
    eid          Dsrc-EID OPTIONAL,
    parameter    ApplicationContextMark OPTIONAL
}

AttributeIdList ::= SEQUENCE (SIZE(0.. 127,...)) OF INTEGER(0..127,...)
AttributeList{Container} ::= SEQUENCE (SIZE(0..127,...)) OF
Attributes{Container}

Attributes{Container} ::= SEQUENCE {
    attributeId    INTEGER (0..127,...),
    attributeValue Container
}

BeaconID ::= SEQUENCE{
    manufacturerid    INTEGER(0.. 65535),
    individualid      INTEGER(0..134217727)
} -- for registration of manufacturerid see www.nni.nl/cen278

BroadcastPool ::= SEQUENCE{
    directoryvalue    Directory,
    content            SEQUENCE (SIZE(0..127,...)) OF File
}

BST ::= SEQUENCE{
    rsu              BeaconID,
    time             Time,
    profile          Profile,
    mandApplications ApplicationList,
    nonmandApplications ApplicationList OPTIONAL,
    profileList      SEQUENCE (SIZE(0..127,...)) OF Profile
}

EfcContainer ::= CHOICE{
-- The alternative for container and its value is determined
-- from the service primitives.
    integer          [0] INTEGER,
    bitString        [1] BIT STRING,
    octetstring      [2] OCTET STRING (SIZE (0..127,...)),
    universalString  [3] UniversalString,
    beaconId         [4] BeaconID,
    t-apdu           [5] T-APDUs,
    dsrcApplicationEntityId [6] DsrcApplicationEntityID,
    dsrc-Ase-Id      [7] Dsrc-EID,
    attrIdList       [8] AttributeIdList,
    attrList         [9] AttributeList{EfcContainer},
    broadcastPool    [10] BroadcastPool,
    directory        [11] Directory,
    file             [12] File,
    fileType         [13] FileType,
    record           [14] Record,
    time            [15] Time,
    vector           [16] SEQUENCE (SIZE(0..255)) OF INTEGER(0..127,...),
    gstrq           [17] GetStampedRq,
    gstrs           [18] GetStampedRs{EfcContainer},
    sstrq           [19] SetStampedRq{EfcContainer},
    ginrq           [20] GetInstanceRq,

```