International Standard

ISO 17573-3

Second edition
2024-12

Electronic fee collection — System architecture for vehicle-related tolling —

Part 3: Data dictionary

Perception de télépéage — Architecture de systèmes pour le péage lié aux véhicules —

Partie 3: Dictionnaire de données

Reference number
ISO 17573-3:2024(en)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

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 document 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).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

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

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems,* in collaboration with the European Committee for Standardization (CEN) Technical Committee CEN/TC 278, *Intelligent transport systems,* in accordance with the Agreement on technical cooperation between ISO and CEN (Vienna Agreement).

This third edition cancels and replaces the second edition (ISO 17573-3:2023), which has been technically revised.

The main changes are as follows:

— the `AlphabetIndicator` in the definition of a licence plate has been corrected to be encoded as a 6-bit value when using unaligned packed encoding rules;

— `AlphabetIndicator`, `DriverClass`, `TripPurpose` and vehicle-dimension-related parameters have been defined as separate data types.

A list of all parts in the ISO 17573 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

This document is a part of the ISO 17573 series which defines the system architecture for vehicle-related tolling. ISO 17573-1 gives a reference model for the system architecture. ISO/TS 17573-2 provides a collection of terms and definitions within the field of electronic fee collection (EFC) and road user charging that are used in the different documents published by ISO and CEN under the general title *Electronic fee collection*.

This document (ISO 17573-3) provides a data dictionary that contains the definitions of ASN.1 (data) types and the associated semantics.

The document is intended to be used as a reference by editors of ISO and CEN documents on EFC and in related areas of standardization (such as Intelligent transport systems, ITS).

It is foreseen that the library of ASN.1 (data) types contained in this document will be augmented with additional definitions as these become available.

Electronic fee collection — System architecture for vehicle-related tolling —

Part 3:
Data dictionary

1 Scope

This document specifies the syntax and semantics of data objects in the field of electronic fee collection (EFC). The definitions of data types and assignment of semantics are provided in accordance with the abstract syntax notation one (ASN.1) technique, as specified in ISO/IEC 8824-1. This document defines:

— ASN.1 (data) types within the field of EFC;

— ASN.1 (data) types of a more general use that are used more specifically in standards related to EFC.

This document does not seek to define ASN.1 (data) types that are primarily related to other fields that operate in conjunction with EFC, such as cooperative intelligent transport systems (C-ITS), the financial sector, etc.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/TS 17573-2, *Electronic fee collection — System architecture for vehicle related tolling — Part 2: Vocabulary*

ISO 612, *Road vehicles — Dimensions of motor vehicles and towed vehicles — Terms and definitions*

ISO 1176, *Road vehicles — Masses — Vocabulary and codes*

ISO 3166-1, *Codes for the representation of names of countries and their subdivisions — Part 1: Country code*

ISO 4217, *Codes for the representation of currencies*

ISO/IEC 7812-1, *Identification cards — Identification of issuers — Part 1: Numbering system*

ISO/IEC 7812-2, *Identification cards — Identification of issuers — Part 2: Application and registration procedures*

ISO/IEC 8859-1, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1*

ISO/IEC 8859-2, *Information technology — 8-bit single-byte coded graphic character sets — Part 2: Latin alphabet No. 2*

ISO/IEC 8859-3, *Information technology — 8-bit single-byte coded graphic character sets — Part 3: Latin alphabet No. 3*

ISO/IEC 8859-4, *Information technology — 8-bit single-byte coded graphic character sets — Part 4: Latin alphabet No. 4*

ISO/IEC 8859-5, *Information technology — 8-bit single-byte coded graphic character sets — Part 5: Latin/Cyrillic alphabet*

ISO/IEC 8859-6, *Information technology — 8-bit single-byte coded graphic character sets — Part 6: Latin/Arabic alphabet*

ISO/IEC 8859-7, *Information technology — 8-bit single-byte coded graphic character sets — Part 7: Latin/Greek alphabet*

ISO/IEC 8859-8, *Information technology — 8-bit single-byte coded graphic character sets — Part 8: Latin/Hebrew alphabet*

ISO/IEC 8859-9, *Information technology — 8-bit single-byte coded graphic character sets — Part 9: Latin alphabet No. 5*

ISO/IEC 8859-10, *Information technology — 8-bit single-byte coded graphic character sets — Part 10: Latin alphabet No. 6*

ISO/IEC 8859-11, *Information technology — 8-bit single-byte coded graphic character sets — Part 11: Latin/Thai alphabet*

ISO/IEC 8859-13, *Information technology — 8-bit single-byte coded graphic character sets — Part 13: Latin alphabet No. 7*

ISO/IEC 8859-14, *Information technology — 8-bit single-byte coded graphic character sets — Part 14: Latin alphabet No. 8 (Celtic)*

ISO/IEC 8859-15, *Information technology — 8-bit single-byte coded graphic character sets — Part 15: Latin alphabet No. 9*

ISO/IEC 8859-16, *Information technology — 8-bit single-byte coded graphic character sets — Part 16: Latin alphabet No. 10*

ISO/IEC 10646, *Information technology — Universal coded character set (UCS)*

ISO/IEC 646, *Information technology — ISO 7-bit coded character set for information interchange*

Indian standard (IS) 13194, *Indian script code for information interchange — ISCII*

Thai Industrial Standard (TIS) 620-2533, *Standard for Thai character codes for computers*

Vietnamese Standard (TCVN) 5712, *Information Technology — Standard 8-bit Vietnamese character code set for use in information exchange*

RFC 1489, *Registration of a Cyrillic Character Set*

RFC 2319, *Ukrainian Character Set KOI8-U*

Japan Industrial Standard (JIS) X 0213, *Japanese standard character set*

Chinese Standard (GB) 2312, *Code of Chinese graphic character set for information interchange — Primary set*

Chinese National Standard (CNS) 11643, *National Chinese standard interchange code*

Korean Standard (KS) X 1001, *Korean national standard for character encoding*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/TS 17573-2 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
BITSTRING
<type> *simple type* (3.14) whose distinguished values are an ordered sequence of zero, one or more bits

[SOURCE: ISO/IEC 8824-1:2021, 3.8.7, modified — Term modified from "BITSTRING type" to "BITSTRING" and domain "<type>" added. NOTE has been removed.]

3.2
CHOICE
<type> type defined by referencing a list of distinct types; each value of the choice type is derived from the value of one of the *component types* (3.4)

Note 1 to entry: Each value of the choice type is derived from the value of one of the component types.

[SOURCE: ISO/IEC 8824-1:2021, 3.8.14, modified — Term modified from "CHOICE type" to "CHOICE" and domain "<type>" added. Note 1 to entry has been added.]

3.3
complex data type
one type that has more than *three levels* (3.17)

3.4
component type
one of the types referenced when defining a `CHOICE` (3.2), `SET` (3.12), `SEQUENCE` (3.10), `SET OF` (3.13), or `SEQUENCE OF` (3.11)

[SOURCE: ISO/IEC 8824-1:2021, 3.8.15]

3.5
data type
categorization of an abstract set of possible values, characteristics, and set of operations for an attribute

[SOURCE: ISO/IEC 25012:2008, 4.7 — modified, Note 1 to entry has been removed.]

3.6
INTEGER
<type> *simple type* (3.14) with distinguished values which are the positive and negative whole numbers, including zero (as a single value)

[SOURCE: ISO/IEC 8824-1:2021, 3.8.48, modified — Term modified from "INTEGER type" to "INTEGER" and domain "<type>" added. NOTE has been removed.]

3.7
object
well-defined piece of information, definition, or specification which requires a name in order to identify its use in an instance of communication

[SOURCE: ISO/IEC 8824-1:2021, 3.8.52, modified — NOTE has been removed.]

3.8
OCTET STRING
<type> *simple type* (3.14) whose distinguished values are an ordered sequence of zero, one or more octets, each octet being an ordered sequence of eight bits

[SOURCE: ISO/IEC 8824-1:2021, 3.8.55, modified — Term modified from "OCTET STRING type" to "OCTET STRING" and domain "<type>" added.]

3.9
parent type
type that is being constrained when defining a *subtype* (3.16), and which governs the subtype notation

[SOURCE: ISO/IEC 8824-1:2021, 3.8.58, modified — Term modified from "parent type (of a subtype)" to "parent type". NOTE has been removed.]

3.10
SEQUENCE
<type> type defined by referencing a fixed, ordered list of types (some of which can be declared to be optional)

Note 1 to entry: Each value of the SEQUENCE type is an ordered list of values, one from each *component type* (3.4).

[SOURCE: ISO/IEC 8824-1:2021, 3.8.67, modified — Term modified from "SEQUENCE types" to "SEQUENCE" and domain "<type>" added. Second part of original definition moved to Note 1 to entry.]

3.11
SEQUENCE-OF
<type> type defined by referencing a single *component type* (3.4)

Note 1 to entry: Each value in the SEQUENCE-OF type is an ordered list of zero, one or more values of the component type.

[SOURCE: ISO/IEC 8824-1:2021, 3.8.68, modified — Term modified from "SEQUENCE-OF types" to "SEQUENCE-OF" and domain "<type>" added. Second part of original definition moved to Note 1 to entry.]

3.12
SET
<type> type defined by referencing a fixed, unordered, list of types (some of which may be declared to be optional)

Note 1 to entry: Each value in the SET type is an unordered list of values, one from each *component type* (3.4).

Note 2 to entry: Where a component type is declared to be optional, a value of the SET type need not contain a value of that component type.

[SOURCE: ISO/IEC 8824-1:2021, 3.8.72, modified — Term modified from "SET types" to "SET" and domain "<type>" added. Second part of original definition moved to Note 1 to entry. Note 1 to entry updated as Note 2 to entry.]

3.13
SET-OF
<type> types defined by referencing a single *component type* (3.4)

Note 1 to entry: Each value in the SET-OF type is an unordered list of zero, one or more values of the component type.

[SOURCE: ISO/IEC 8824-1:2021, 3.8.73, modified — Term modified from "SET-OF types" to "SET-OF" and domain "<type>" added. Second part of original definition moved to Note 1 to entry.]

3.14
simple type
type defined by directly specifying the set of their values

[SOURCE: ISO/IEC 8824-1:2021, 3.8.74]

3.15
single-level data type
data type (3.5) which is a `SEQUENCE` (3.10), or `SEQUENCE OF` (3.11) defined by referencing a *simple type* (3.14) or a *subtype* (3.16) of a simple type

3.16
subtype
<parent type> type whose values are a subset (or the complete set) of the values of some other type *(*the *parent type)* (3.9)

[SOURCE: ISO/IEC 8824-1:2021, 3.8.76, modified — Term modified from "subtype (of a parent type)" to "subtype" and domain "<parent type>" added.]

3.17
three-level data type
data type (3.5) which is a `CHOICE` (3.2), `SEQUENCE` (3.10), or `SEQUENCE OF` (3.11) defined by referencing at least one *two-level data type* (3.18)

3.18
two-level data type
data type (3.5) which is a `CHOICE` (3.2), `SEQUENCE` (3.10), or `SEQUENCE OF` (3.11) defined by referencing *a single-level data type* (3.15)

4 Abbreviated terms

ASN.1	abstract syntax notation one
BCD	binary coded decimal
CO	carbon monoxide
CO_2	carbon dioxide
DSRC	dedicated short-range communication
EFC	electronic fee collection
GLONASS	global navigation satellite system of the Russian Federation
GNSS	global navigation satellite system
GPS	global positioning system
GTRF	global terrestrial reference system
HC	hydrocarbon
ICC	integrated circuit(s) card
ITRF	international terrestrial reference frame
ITRS	international terrestrial reference system
JGS	Japan satellite navigation geodetic system
LAC	localization augmentation communication
NO_X	nitrogen oxides
OBE	on-board equipment
OBU	on-board unit
QZSS	quasi-zenith satellite system
RSE	roadside equipment

TC	toll charger
TSP	toll service provider
UCS	universal character set
VAT	value-added tax

5 EFC common data object definitions

5.1 General

In this clause, the structure of all EFC common data objects is described. The formal definition provided in Annex A in terms of data type definitions applies. In addition to the structure description, each data object is also given a semantics.

Each one of the common data types defined herein is used by more than one document in the suite of CEN or ISO documents in the field of EFC. These documents may also define their own data types when none of the common data types defined herein satisfies their need.

The definitions of the EFC common data types are ordered according to their data type level:

— first subtypes based on simple data types (e.g. INTEGER or OCTET STRING),

— then single-level data types,

— then two-level data types,

— then three-level data types,

— then complex data types.

Data types are ordered alphabetically within each level.

5.2 Subtypes of simple data types

5.2.1 AccountStatus

The data type `AccountStatus` shall be of the simple type as specified in Table 1.

Table 1 — `AccountStatus`

Subtype	Parent type	Semantics
-	`INTEGER`	`AccountStatus` provides the status of the user's account. The following semantics are assigned: — `ok`: The amount stored in the account is above or equal to the predefined threshold; — `low`: The amount stored in the account has fallen below a predefined threshold; — `empty`: The amount stored in the account is zero; — `negative`: The amount stored in the account has fallen below zero.

5.2.2 ActualNumberOfPassengers

The data type `ActualNumberOfPassengers` shall be of the subtype as specified in Table 2.

Table 2 — `ActualNumberOfPassengers`

Subtype	Parent type	Semantics
`Int1Unsigned`		`ActualNumberOfPassengers` represents the actual number of passengers (i.e. persons) present in the vehicle, including the driver. This information can affect the applicability of tolls or the value of the tariff to be applied, e.g. in High Occupancy Tolling or High Occupancy Vehicle lanes.

5.2.3 AlphabetIndicator

The data type `AlphabetIndicator` shall be of the simple type as specified in Table 3.

Table 3 — `AlphabetIndicator`

Subtype	Parent type	Semantics
-	`ENUMERATED`	`AlphabetIndicator` indicates the type of alphabet used to represent the licence plate number. 7-bit character-sets (e.g. ascii or iscii) are encoded. NOTE ISO 14906:2022, Annex D, provides a mapping from `latinAlphabetNo5` to `latinAlphabetNo1`.

The following semantics have been assigned to the values of `AlphabetIndicator`:

— `latinAlphabetNo1`: Latin-1 in accordance with ISO/IEC 8859-1,

— `latinAlphabetNo2`: Latin-2 in accordance with ISO/IEC 8859-2,

— `latinAlphabetNo3`: latin-3 in accordance with ISO/IEC 8859-3,

— `latinAlphabetNo4`: Latin-4 in accordance with ISO/IEC 8859-4,

— `latinCyrillicAlphabet`: Cyrillic in accordance with ISO/IEC 8859-5,

— `latinArabicAlphabet`: Arabic in accordance with ISO/IEC 8859-6,

— `latinGreekAlphabet`: Greek in accordance with ISO/IEC 8859-7,

— `latinHebrewAlphabet`: Hebrew in accordance with ISO/IEC 8859-8,

— `latinAlphabetNo5`: Latin-5 in accordance with ISO/IEC 8859-9,

— `latinAlphabetNo6`: Latin-6 in accordance with ISO/IEC 8859-10,

— `twoOctetBMP`: shall not be used,

— `forOctetCannonical`: shall not be used,

— `latinAlphabetNo7`: Latin-7 in accordance with ISO/IEC 8859-13,

— `latinAlphabetNo8`: Latin-8 in accordance with ISO/IEC 8859-14,

— `latinAlphabetNo9`: Latin-9 in accordance with ISO/IEC 8859-15,

— `latinAlphabetNo10`: Latin-10 in accordance with ISO/IEC 8859-16,

— `latinThaiAlphabet`: Thai in accordance with ISO/IEC 8859-11,

— `utf8`: UCS transformation format with a variable length of 1 byte to 4 bytes per character in accordance with ISO/IEC 10646 and specified in RFC 3629,

— `utf16BigEndian`: UCS transformation format with a length of 2 bytes or 4 bytes per character in accordance with ISO/IEC 10646 and specified in RFC 2781,

— `utf16LittleEndian`: UCS transformation format with a length of 2 bytes or 4 bytes per character in accordance with ISO/IEC 10646 and specified in RFC 2781,

— `utf32BigEndian`: UCS transformation format with a fixed length of 4 bytes per character in accordance with ISO/IEC 10646,

— `utf32LittleEndian`: UCS transformation format with a fixed length of 4 bytes per character in accordance with ISO/IEC 10646,

— `ucs4`: Universal character set with a length of 2 or 4 bytes per character in accordance with ISO/IEC 10646,

— `ascii`: American standard code for information interchange in accordance with ISO/IEC 646,

— `iscii`: Indian character set with a length of 1 byte per character in accordance with IS 13194,

— `tis620`: Thai character set with a length of 1 byte per character in accordance with Thai Industrial Standard TIS 620-2533,

— `vscii`: Vietnamese character set with a length of 1 byte per character in accordance with TCVN 5712,

— `koi8R`: Russian character set with a length of 1 byte per character in accordance with RFC 1489,

— `koi8U`: Ukrainian character set with a length of 1 byte per character in accordance with RFC 2319,

— `jisX0213`: Japanese character set with a length of 2 byte per character in accordance with JIS X 0213,

— `gbt2312`: Simplified Chinese character set with a length of 2 byte per character in accordance with GB 2312,

— `big5`: Traditional Chinese character set with a length of 2 byte per character in accordance with CNS 11643,

— `ksx1001`: Korean character set with a length of 2 byte per character in accordance with KS X 1001.

5.2.4 Altitude

The data type `Altitude` shall be of the simple type as specified in Table 4.

Table 4 — `Altitude`

Subtype	Parent type	Semantics
`Int2Signed`		`Altitude` provides the ellipsoidal height (in units of 0,25 m) above or below the WGS84[5] ellipsoid of the geographical point. The range in m is from -8 192,00 to +8 191,75. NOTE WGS84[9] represents a broadly adopted global geodetic reference system for the Earth for practical applications of mapping, geo-positioning and navigation. Other terrestrial reference frames exist, notably the International Terrestrial Reference Frame (ITRF, the latest edition currently being ITRF 2020). It is possible to convert between the most commonly used terrestrial reference frames and the differences between them are typically in the order of centimetres. The international terrestrial reference frame is becoming increasingly recognized and used as the primary reference frame. All recent and up-to-date Global Navigation Satellite System (GNSS)-specific terrestrial reference frames (WGS 84 for GPS, PZ-90 for GLONASS, the GTRF for Galileo, CGCS2000 for BeiDou, and the JGS for QZSS) are aligned to a primary ITRS[10] realization, according to ISO 19161-1:2020, Annex C.

5.2.5 Axles

The data type `Axles` shall be of the simple type as specified in Table 5.

Table 5 — `Axles`

Subtype	Parent type	Semantics
-	INTEGER	`Axles` provides the number of axles of either the tractor or trailer including drop axles.

5.2.6 CabType

The data type `CabType` shall be of the simple constrained type as specified in Table 6.

Table 6 — `CabType`

Subtype	Parent type	Semantics
-	INTEGER	`CabType` provides information about the cabin type of a vehicle according to Regulation (EU) 2019/1242,[17] Annex I. The following semantics are assigned: — `dayCab`: A type of cabin that is not a sleeper cabin; — `sleeperCab`: A type of cabin that has a compartment behind the driver's seat intended to be used for sleeping as reported in accordance with Regulation (EU) 2018/956,[19] Annex I, Part B, Data no. 84.

5.2.7 ChassisType

The data type `ChassisType` shall be of the simple constrained type as specified in Table 7.

Table 7 — `ChassisType`

Subtype	Parent type	Semantics
-	INTEGER	`ChassisType` provides information about the type of chassis of a vehicle according to Regulation (EU) 2019/1242 and Regulation (EU) 2018/858.[18] The following semantics are assigned: — `rigidLorry`: A type of chassis, that is not designed or constructed for the towing of a semitrailer, as defined in Regulation (EU) 2019/1242,[17] Article 3 (6); — `tractor`: A type of chassis, that is designed and constructed exclusively or principally to tow semitrailers, as defined in Regulation (EU) 2019/1242,[17] Article 3 (7); — `van`: A lorry with the compartment where the driver is located and cargo area within a single unit, as defined in Regulation (EU) 2018/858,[18] Annex I, Part C, 4.2.

5.2.8 Co2EmissionClass

The data type `Co2EmissionClass` shall be of the simple type as specified in Table 8.

Table 8 — `Co2EmissionClass`

Subtype	Parent type	Semantics
-	`INTEGER`	`Co2EmissionClass` indicates the CO_2 emission class according to the selected CO_2 scheme. The following semantics are assigned: — `noEntry`: No CO_2 emission class assigned; — `co2EmissionClass1`: CO_2 emission class 1; — `co2EmissionClass2`: CO_2 emission class 2; — `co2EmissionClass3`: CO_2 emission class 3; — `co2EmissionClass4`: CO_2 emission class 4; — `co2EmissionClass5`: CO_2 emission class 5; — `co2EmissionClass6`: CO_2 emission class 6; — `co2EmissionClass7`: CO_2 emission class 7.

EXAMPLE The values of `Co2EmissionClass` for heavy-duty vehicles have the meaning as specified in Directive 1999/62/EC,[14] Article 7ga when the `Co2Scheme` is set to `eU`:

— `co2EmissionClass1`, indicating the CO_2 emission class 1 (vehicles that do not belong to any of the other CO_2 emission classes);

— `co2EmissionClass2`, indicating the CO_2 emission class 2 (vehicles with CO_2 emissions more than 5 % below the emission reduction trajectory);

— `co2EmissionClass3`, indicating the CO_2 emission class 3 (vehicles with CO_2 emissions more than 8 % below the emission reduction trajectory);

— `co2EmissionClass4`, indicating the CO_2 emission class 4 (low-emission heavy-duty vehicles);

— `co2EmissionClass5`, indicating the CO_2 emission class 5 (zero emission vehicles).

5.2.9 Co2EmissionValue

The data type `Co2EmissionValue` shall be of the subtype as specified in Table 9.

Table 9 — `Co2EmissionValue`

Subtype	Parent type	Semantics
`Int2Unsigned`		`Co2EmissionValue` represents the vehicle's CO_2 emission value according to vehicle registration documents, in g/km.

5.2.10 Co2EmissionValueLoad

The data type `Co2EmissionValueLoad` shall be of the subtype as specified in Table 10.

Table 10 — `Co2EmissionValueLoad`

Subtype	Parent type	Semantics
`Int4Unsigned`		`Co2EmissionValueLoad` represents the vehicle's CO_2 emission value according to vehicle registration documents, in 0,01 g/tkm steps. EXAMPLE A value of 10 000 represents a CO_2 emission of 100 g/tkm.

5.2.11 Co2Scheme

The data type `Co2Scheme` shall be of the simple type as specified in Table 11.

Table 11 — `Co2Scheme`

Subtype	Parent type	Semantics
-	`INTEGER`	`Co2Scheme` indicates the CO_2 emission class scheme applied to the values defined in `Co2EmissionClass`. The following semantics are assigned: — `noEntry`: No CO_2 scheme applied; — `eU`: CO_2 scheme according to Directive 1999/62/EC,[15] Article 7ga for heavy-duty vehicles and Article 7gb for light-duty vehicles.

5.2.12 ContractAuthenticator

The data type `ContractAuthenticator` shall be of the simple type as specified in Table 12.

Table 12 — `ContractAuthenticator`

Subtype	Parent type	Semantics
-	`OCTET STRING`	`ContractAuthenticator` is an authenticator calculated by the toll service provider (TSP) when issuing the contract, to prevent tampering with contract data.

5.2.13 ContractSerialNumber

The data type `ContractSerialNumber` shall be of the subtype as specified in Table 13.

Table 13 — `ContractSerialNumber`

Subtype	Parent type	Semantics
`Int4Unsigned`		`ContractSerialNumber` is an integer designating the individual contract, assigned at the discretion of the TSP.

5.2.14 CopValue

The data type `CopValue` shall be of the simple type as specified in Table 14.

Table 14 — `CopValue`

Subtype	Parent type	Semantics
-	`INTEGER`	`CopValue` represents the vehicle's carbon dioxide emission values expressed in g/km. The following semantics are assigned: — `noEntry`: Value not defined; — `co2Class1`: For emission values below 101 g/km; — `co2Class2`: For emission value from 101 g/km to 120 g/km; — `co2Class3`: For emission values from 121 g/km to 140 g/km; — `co2Class4`: For emission values from 141 g/km to 160 g/km; — `co2Class5`: For emission values from 161 g/km to 200 g/km; — `co2Class6`: For emission values from 201 g/km to 250 g/km; — `co2Class7`: For emission values above 250 g/km.

NOTE The carbon dioxide emission value of a vehicle is stated in field V.7 of the vehicle's European registration certificate, according to Directive 2003/127/EC.[15]

5.2.15 CountryCode

The data type `CountryCode` shall be of the simple type as specified in Table 15.

Table 15 — `CountryCode`

Subtype	Parent type	Semantics
-	`BITSTRING`	`CountryCode` represents an ISO 3166-1 country code. Values are encoded in accordance with the ITA-2 encoding of the ISO 3166-1 country code. EXAMPLE 1 Austria (AT) = 11000 00001. EXAMPLE 2 Belgium (BE) = 10011 10000.

5.2.16 DetectionMode

The data type `DetectionMode` shall be of the simple type as specified in Table 16.

Table 16 — `DetectionMode`

Subtype	Parent type	Semantics
-	`INTEGER`	`DetectionMode` indicates how the charge object was detected. The following semantics are assigned: — `measured`: The charge object was detected by evaluation of positioning data using regular rules defined for recognizing charge objects; — `inferred`: The charge object was not detected by evaluation of positioning data, but was inferred from the overall trip logic; — `lac`: For implementation reasons in special cases, the normal charge object detection technology can be supported by localization augmentation communication (LAC) beacon, which communicates its location, usually using dedicated short-range communication (DSRC) technology.

5.2.17 DistanceUnit

The data type `DistanceUnit` shall be of the simple type as specified in Table 17.

Table 17 — `DistanceUnit`

Subtype	Parent type	Semantics
-	INTEGER	`DistanceUnit` identifies units of distance. The following semantics are assigned: — `kilometres`, — `metres`, — `decimetres`, — `centimetres`, — `millimetres`. Values identifying non-ISO units are maintained in the ASN.1 definition for backward compatibility, although deprecated.

5.2.18 DriverClass

The data type `DriverClass` shall be of the simple type as specified in Table 18.

Table 18 — `DriverClass`

Subtype	Parent type	Semantics
-	INTEGER	`DriverClass` provides a description of the driver's characteristics as pertinent to the calculation of the tariff, contract provider specific coding: — `genericPerson`: Indicating a generic person; — `reducedMobility`: Indicating a person with reduced mobility; — `otherDisability`: Indicating a person with a disability other than reduced mobility (e.g. impaired hearing); — `officialAuthority`: Indicating an official authority; — `notToBeDisclosed`: Indicating that the class of the driver is not disclosed; — `craftsPerson`: Indicating a crafts person.

5.2.19 EmissionUnit

The data type `EmissionUnit` shall be of the simple constrained type as specified in Table 19.

Table 19 — `EmissionUnit`

Subtype	Parent type	Semantics
-	INTEGER	`EmissionUnit` represents the unit of emission values as an integer on one octet. The following semantics are assigned: — `mgPerKm`: mg/km, indicating the emission values are expressed in milligrams per kilometre, — `mgPerKwh`: mg/kWh, indicating the emission values are expressed in milligrams per Kilowatt-hour.

5.2.20 EngineCapacity

The data type `EngineCapacity` shall be of the subtype as specified in Table 20.

Table 20 — `EngineCapacity`

Subtype	Parent type	Semantics
`Int2Unsigned`		`EngineCapacity` represents the capacity of the vehicle's engine in cm^3. NOTE In Europe `EngineCapacity` corresponds to "P.1" (capacity) as defined in the vehicle registration documents, according to Commission Directive 1999/37/EC.[13]

5.2.21 EngineCharacteristics

The data type `EngineCharacteristics` shall be of the simple type as specified in Table 21.

Table 21 — `EngineCharacteristics`

Subtype	Parent type	Semantics
-	`INTEGER`	`EngineCharacteristics` provides information about the vehicle's engine type and fuel characteristics.

The following semantics have been assigned:

— `noEntry`, indicating that no information is available;

— `noEngine`, indicating that the vehicle has no engine;

— `petrolUnleaded`, indicating engine operating with unleaded petrol;

— `petrolLeaded`, indicating engine operating with leaded petrol;

— `diesel`, indicating engine operating with diesel;

— `lpg`, indicating engine operating with liquefied petroleum gas;

— `battery`, indicating vehicle powered exclusively by battery;

— `solar`, indicating engine operating with solar energy;

— `hybrid`, value kept for legacy compatibility (more differentiated values are available);

— `hydrogen`, indicating engine operating with hydrogen;

— `multiFuel`, indicating multi fuel engine;

— `bivalentPetrolLpg`, indicating bivalent operating engine with petrol or liquefied petroleum gas;

— `bivalentPetrolCng`, indicating bivalent operating engine with petrol or compressed natural gas;

— `combinedPetrolElectric`, indicating combined operation with petrol and electric engine;

— `cng`, indicating engine operating with compressed natural gas;

— `lng`, indicating engine operating with liquefied natural gas;

— `combinedDieselElectric`, indicating combined operation of diesel and electric engine;

— `combinedHydrogenElectric`, indicating combined operation of hydrogen and electric engine;

— `bivalentHydrogenPetrol`, indicating bivalent operating engine with hydrogen or petrol;

- `bivalentHydrogenPetrolElectricEngine`, indicating bivalent operating engine with hydrogen or petrol combined with electric engine;
- `fuelCellHydrogen`, indicating fuel cell with hydrogen as primary energy source and electric engine;
- `fuelCellPetrol`, indicating fuel cell with petrol as primary energy source and electric engine;
- `fuelCellMethanol`, indicating fuel cell with methanol as primary energy source and electric engine;
- `fuelCellEthanol`, indicating fuel cell with ethanol as primary energy source and electric engine;
- `fuelCellDiesel`, indicating fuel cell with diesel as primary energy source and electric engine;
- `combinedMultiFuelElectricEngine`, indicating combined operation of multi fuel and electric engine;
- `combinedCngElectricEngine`, indicating combined operation with compressed natural gas and electric engine;
- `combinedLngElectricEngine`, indicating combined operation with liquefied natural gas and electric engine;
- `petrolEthanol`, indicating fuel mix of petrol and mainly ethanol, e.g. E85;
- `combinedLpgElectricEngine`, indicating combined operation of liquefied petroleum gas and electric engine;
- `hybridPetrolExternalBattery`, indicating hybrid drive with petrol and external rechargeable battery (plug-in hybrid);
- `hybridDieselExternalBattery`, indicating hybrid drive with diesel and external rechargeable battery (plug-in hybrid);
- `hybridLpgExternalBattery`, indicating hybrid drive with liquefied petroleum gas and external rechargeable battery (plug-in hybrid);
- `hybridHydrogenExternalBattery`, indicating hybrid drive with hydrogen and external rechargeable battery (plug-in hybrid);
- `hybridMultiFuelExternalBattery`, indicating hybrid drive with multi-fuel and external rechargeable battery (plug-in hybrid);
- `hybridCngExternalBattery`, indicating hybrid drive with compressed natural gas and external rechargeable battery (plug-in hybrid);
- `hybridLngExternalBattery`, indicating hybrid drive with liquefied natural gas and external rechargeable battery (plug-in hybrid);
- `hybridBivalentHydrogenPetrolExternalBattery`, indicating hybrid drive with bivalent operating hydrogen and petrol engine and external rechargeable battery (plug-in hybrid);
- `hydrogenCng`, indicating a fuel mix of hydrogen and compressed natural gas;
- `hydrogenLng`, indicating a fuel mix of hydrogen and liquefied natural gas;
- `hybridHydrogenCngExternalBattery`, indicating hybrid drive with hydrogen and compressed natural gas and external chargeable battery (plug-in hybrid);
- `hybridHydrogenLngExternalBattery`, indicating hybrid drive with hydrogen and liquefied natural gas and external chargeable battery (plug-in hybrid);
- `ethanol`, indicating ethanol or fuel mix of ethanol and other fuel (except petrol) or additives, e.g. E95;
- `hybridFuelCellHydrogen`, indicating hybrid drive with fuel cell (electric engine) and hydrogen (combustion engine);
- `hybridFuelCellHydrogenExternalBattery`, indicating hybrid drive with fuel cell (electric engine) and hydrogen (combustion engine) and external chargeable battery (plug-in hybrid);

— `dualFuelLngDiesel`, indicating dual operation with liquefied natural gas and diesel;

— `electricExternal`, indicating electric engine with external power supply;

— `biogas`, indicating mixture of different gases produced by the breakdown of organic matter;

— `bioDiesel`, indicating vegetable oil- or animal fat-based diesel fuel;

— `bioPetrol`, indicating petrol fully or partly based on vegetable sources;

— `bivalentPetrolBiogas`, indicating bivalent operating engine with petrol or biogas;

— `combinedBiogasElectricEngine`, indicating combined operation of biogas and electric engine;

— `dualFuelCngDiesel`, indicating dual operation with compressed natural gas and diesel;

— `other`.

5.2.22 EnginePower

The data type `EnginePower` shall be of the subtype as specified in Table 22.

Table 22 — `EnginePower`

Subtype	Parent type	Semantics
`Int2Unsigned`		`EnginePower` represents the maximum net power of the vehicle's engine in kW. NOTE In Europe `EnginePower` corresponds to "P.2" (maximum net power) as defined in the vehicle registration documents, according to Commission Directive 1999/37/EC.[13]

5.2.23 EquipmentIccId

The data type `EquipmentIccId` shall be of the simple type as specified in Table 23.

Table 23 — `EquipmentIccId`

Subtype	Parent type	Semantics
-	`OCTET STRING`	`EquipmentIccId` represents the identification number of the integrated circuit(s) card (ICC).[6]

5.2.24 EquipmentObuId

The data type `EquipmentObuId` shall be of the simple type as specified in Table 24.

Table 24 — `EquipmentObuId`

Subtype	Parent type	Semantics
-	`OCTET STRING`	`EquipmentObuId` represents the unique identification of the on-board unit (OBU) within the context of the associated manufacturer. NOTE As an example of usage, EN 15509 expresses this value using four octets. Depending on the encoding rules used, an additional octet indicating the length can be inserted.

5.2.25 EquipmentStatus

The data type `EquipmentStatus` shall be of the constrained simple type as specified in Table 25.

Table 25 — `EquipmentStatus`

Subtype	Parent type	Semantics
-	`BITSTRING`	`EquipmentStatus` provides operator-specific EFC application-related information pertaining to the status of the equipment.

5.2.26 EuroValue

The data type `EuroValue` shall be of the simple type as specified in Table 26.

Table 26 — `EuroValue`

Subtype	Parent type	Semantics
-	`INTEGER` NOTE This was changed from data type `ENUMERATED` in ISO 14906:2018 to data type `INTEGER`.	`EuroValue` represents the vehicle's Euro emission category as defined in EU Regulation 2024/1257.[12] The following semantics are assigned: — `noEntry`: Indicating that no Euro emission category value is provided; — `euro1`: Indicating Euro emission category 1; — `euro2`: Indicating Euro emission category 2; — `euro3`: Indicating Euro emission category 3; — `euro4`: Indicating Euro emission category 4; — `euro5`: Indicating Euro emission category 5; — `euro6`: Indicating Euro emission category 6; — `euro7`: Indicating Euro emission category 7; — `eev`: Indicating Enhanced Environmentally Friendly Vehicle class.

5.2.27 EuroValueSubClass

The data type `EuroValueSubClass` shall be of the simple type as specified in Table 27.

Table 27 — `EuroValueSubClass`

Subtype	Parent type	Semantics
-	`INTEGER`	`EuroValueSubClass` represents the vehicle's sub-category which is defined for certain Euro emission categories as defined in EU Regulation 2024/1257.[12] The following semantics are assigned: — `euro7G`: Indicating a subgroup for vehicles equipped with internal combustion engines with geofencing technologies. NOTE These vehicles are equipped with a driver warning system to inform the user when the traction batteries are nearly empty and to stop the vehicle if not charged within 5 km from the first warning while on zero-emission mode inside the geofencing area; — `euro7ext`: Indicating a subgroup for vehicles of category N2 between 3,5 t and 5 t maximum mass originating from a type of vehicle of category N1 if the vehicle meets the requirements for a type of vehicle of category N1; — `euro7Gext`: Indicating a subgroup for vehicles that meet both the characteristics of `euro7G` and `euro7ext`.

5.2.28 IssuerIdentifier

The data type `IssuerIdentifier` shall be of the simple type as specified in Table 28.

Table 28 — `IssuerIdentifier`

Subtype	Parent type	Semantics
-	`INTEGER`	`IssuerIdentifier` provides the identifier of an organization according to the corresponding registry. NOTE See Table 92 for an example of a specific registry usage.

5.2.29 Latitude

The data type `Latitude` shall be of the subtype as specified in Table 29.

Table 29 — `Latitude`

Subtype	Parent type	Semantics
`Int4Signed`		`Latitude` provides the latitude (in micro degrees) of the geographical point, with a range of 90° in the north or south hemisphere. Positive values are used for latitude north of the Equator, and negative values are used for latitude south of the Equator. So, the effective range of `Latitude` is (–90000000 .. 90000000).

5.2.30 LocalVehicleClassId

The data type `LocalVehicleClassId` shall be of the subtype as specified in Table 30.

Table 30 — `LocalVehicleClassId`

Subtype	Parent type	Semantics
`Int2Unsigned`		`LocalVehicleClassId` provides the unique identifier of the local vehicle class.

5.2.31 LocationClassId

The data type `LocationClassId` shall be of the subtype as specified in Table 31.

Table 31 — `LocationClassId`

Subtype	Parent type	Semantics
`Int4Unsigned`		`LocationClassId` provides the unique identifier of the location class.

5.2.32 Longitude

The data type `Longitude` shall be of the subtype as specified in Table 32.

Table 32 — `Longitude`

Subtype	Parent type	Semantics
`Int4Signed`		`Longitude` provides the longitude (in micro degrees) of the geographical point, providing a range of 180° to the east or to the west of the prime meridian (which passes through the Royal Observatory, Greenwich, England). Negative values are used for longitudes to the west, and positive values are used for longitudes to the east. So, the effective range of `Longitude` is (–179999999..180000000).

5.2.33 Month

The data type `Month` shall be of the simple type as specified in Table 33.

Table 33 — `Month`

Subtype	Parent type	Semantics
-	INTEGER	`Month` provides the month of a date in the Gregorian calendar. The following semantics are assigned: — `noMonth (0)`: This value is only allowed to provide no date in the data type `DateCompact` for the use in the data types `ContractValidity`, `PaymentMeans` and `ValidityOfContract`; — `january (1)`; — `february (2)`; — `march (3)`; — `april (4)`; — `may (5)`; — `june (6)`; — `july (7)`; — `august (8)`; — `september (9)`; — `october (10)`; — `november (11)`; — `december (12)`;

5.2.34 PaymentSecurityData

The data type `PaymentSecurityData` shall be of the simple type as specified in Table 34.

Table 34 — `PaymentSecurityData`

Subtype	Parent type	Semantics
-	OCTET STRING	`PaymentSecurityData` provides security-related data for the authentication of the data integrity.

5.2.35 PayUnit

The data type `PayUnit` shall be of the simple constrained type as specified in Table 35.

Table 35 — `PayUnit`

Subtype	Parent type	Semantics
-	`OCTET STRING`	`PayUnit` represents the unique designation of a factor to express a payment means value in multiples or fractions of a value and a currency in accordance with ISO 4217. The first hexadecimal character (4 bit) expresses a company-specific token or a "charging unit code" as used in the unit in which the fee is expressed. The following semantics are assigned: — `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 1 000 :1; — `4xxx`'H: Currency in 'major' units / 10 (e.g. EUR 10); — `5xxx`'H: Currency in 'major' units / 100 (e.g. USD 100); — `6xxx`'H: Currency in 'major' units / 1 000; — `7xxx`'H: Currency in 'major' units / 10 000; — `8xxx`'H: Currency in 'major' units / 100 000; — `9xxx`'H: Currency in minor units of 10 000 :1; — `Axxx`'H: Currency in minor units of 100 000 :1; — `Bxxx`'H: Currency in minor units of 1 000 000 :1; — `Cxxx`'H: Tokens, specifying a Purse Provider specific coding; — `Dxxx`'H: Charging Unit Codes denoting quantification of the service provided (e.g. worker-hours). The last 3 hexadecimal characters (12 bits) is the binary coded decimal (BCD) representation of the numerical currency code in accordance with the definition in ISO 4217. EXAMPLE A `PayUnit` value of $2\,978_h$ means that an amount is stated in Euro cents.

5.2.36 PersonalAccountNumber

The data type `PersonalAccountNumber` shall be of the simple type as specified in Table 36.

Table 36 — `PersonalAccountNumber`

Subtype	Parent type	Semantics
-	`OCTET STRING`	`PersonalAccountNumber` represents the personal account number structure in accordance with ISO/IEC 7812-1 and ISO/IEC 7812-2 using BCD sub-structures.

5.2.37 RearWheelsSteeringType

The data type `RearWheelsSteeringType` shall be of the simple type as specified in Table 37.

Table 37 — `RearWheelsSteeringType`

Subtype	Parent type	Semantics
-	`INTEGER`	`RearWheelsSteeringType` represents the type of steering for rear wheels. The following semantics are assigned: — `noRearWheels`: None of the rear wheels are steered; — `foreMostRearWheels`: The foremost of the rear wheels are steered; — `rearMostRearWheels`: The rearmost of the rear wheels are steered; — `allRearWheels`: All of the rear wheels are steered.

5.2.38 ReceiptAuthenticator

The data type `ReceiptAuthenticator` shall be of the simple type as specified in Table 38.

Table 38 — `ReceiptAuthenticator`

Subtype	Parent type	Semantics
-	`OCTET STRING`	`ReceiptAuthenticator` contains an Authenticator over some attributes of the data group Receipt, calculated by the session service provider.

5.2.39 ReceiptDistance

The data type `ReceiptDistance` shall be of the subtype as specified in Table 39.

Table 39 — `ReceiptDistance`

Subtype	Parent type	Semantics
`Int3Unsigned`		`ReceiptDistance` contains the total distance covered by the vehicle, since the beginning of its existence, in units of 100 m expressed as an integer.

5.2.40 ResultFin

The data type `ResultFin` shall be of the simple constrained type as specified in Table 40.

Table 40 — `ResultFin`

Subtype	Parent type	Semantics
-	`OCTET STRING`	`ResultFin` represents in one octet the operational result of the financial transaction. The following semantics are assigned: — Most significant 4 bits: 0 means OK: — '0x'H OK; — Most significant 4 bits > 0 means 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; — Least significant 4 bits mean: — '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.

5.2.41 ReceiptIccId

The data type `ReceiptIccId` shall be of the simple type as specified in Table 41.

Table 41 — `ReceiptIccId`

Subtype	Parent type	Semantics
-	`OCTET STRING`	`ReceiptIccId` provides the identification number of the ICC used in the session.

5.2.42 ReceiptObuId

The data type `ReceiptObuId` shall be of the simple type as specified in Table 42.

Table 42 — `ReceiptObuId`

Subtype	Parent type	Semantics
-	`OCTET STRING`	`ReceiptObuId` represents the serial number of the OBU used in the session, unique within the context of the manufacturer.

NOTE This type is only maintained for backward compatibility reasons. Its usage is deprecated.

5.2.43 ResultOp

The data type `ResultOp` shall be of the simple type as specified in Table 43.

Table 43 — `ResultOp`

Subtype	Parent type	Semantics
-	`INTEGER`	`ResultOp` represents the operational result of the EFC session. The following semantics are assigned: — `correctTransaction`, the transaction has been accepted; — `obeStatusNotAccepted`, not acceptable OBE status; — `equipmentStatusNotAccepted`, not acceptable equipmentStatus; — `contractNotInAccessList`, contract shall be in access list; NOTE 1 The deprecated term `contractIdentifierInWhiteList` is no longer used. — `contractIdentifierInBlockList`, contract block listed; NOTE 2 The deprecated term `contractIdentifierInBlackList` is no longer used. — `contractIdentifierNotCorrect`, contract identifier not valid; — `expiredContract`, contract expired; — `contractRestrictionsNotFulfilled`, restrictions in the contract not fulfilled; — `claimedVehicleCharacteristicsNotValid`, vehicle characteristics not conforming to those detected; — `vehicleClassAuthenticationFailed`, failure in vehicle class authentication; — `entryVehicleClassDifferentFromExitVehicleClass`, vehicle class in exit not conforming to vehicle class in entry; — `entryReceiptMissing`, missing proof of entry; — `entryReceiptNotValid`, invalid proof of entry; — `entryTollStationNotValid`, invalid entry station; — `equipmentNotCertified`, equipment not certified; — `timeDifference`, problem with the time difference of the two last receipts; — `accessCredentialsNotAccepted`, invalid access credentials; — `contractAuthenticatorNotAccepted`, invalid contract authenticator; — `receiptAuthenticatorNotAccepted`, invalid receipt authenticator;

Table 43 *(continued)*

Subtype	Parent type	Semantics
		— `claimedVehicleCharacteristicsMissing`, vehicle characteristics not detected; — `paymentMeansNotAccepted`, payment means not accepted; — `paymentAuthenticatorNotAccepted`, invalid payment authenticator; — `paymentMeansInBlockList`, payment means blocklisted; NOTE 3 The deprecated term `contractIdentifierInBlackList` is no longer used. — `paymentMeansNotCorrect`, incorrect payment means; — `expiredPaymentMeans`, expired payment means; — `paymentMeansRestrictionsNotFulfilled`, restrictions in payment means not fulfilled.

5.2.44 ReceiptServiceSerialNumber

The data type `ReceiptServiceSerialNumber` shall be of the subtype as specified in Table 44.

Table 44 — `ReceiptServiceSerialNumber`

Subtype	Parent type	Semantics
`Int3Unsigned`		`ReceiptServiceSerialNumber` represents a specific serial number of the EFC session. NOTE This number can be given, for example, by a dedicated short-range communication (DSRC)-based toll system, or roadside equipment (RSE).

5.2.45 ReceiptText

The data type `ReceiptText` shall be of the simple type as specified in Table 45.

Table 45 — `ReceiptText`

Subtype	Parent type	Semantics
-	`OCTET STRING`	`ReceiptText` contains plain text decodable by the on-board equipment (OBE).

5.2.46 StationType

The data type `StationType` shall be of the simple type as specified in Table 46.

Table 46 — `StationType`

Subtype	Parent type	Semantics
-	`INTEGER`	`StationType` represents the type of EFC Station. The following semantics are assigned: — `unspecified`: No type is specified; — `closedEntryWithPayment`: A payment entry station in a closed system; — `closedEntryWithoutPayment`: An entry station in a closed system without payment; — `closedTransit`: A transit station in a closed system; — `closedExit`: An exit station in a closed system; — `closedCredit`: A station where payment is by means of a rechargeable OBE; — `mixed`: A multi-purpose station; — `passage`: An exit station in an open system; — `checkpoint`: A checkpoint station; — `reload`: A station where a rechargeable OBE can be recharged.

5.2.47 SuspensionType

The data type `SuspensionType` shall be of the simple type as specified in Table 47.

Table 47 — `SuspensionType`

Subtype	Parent type	Semantics
-	`INTEGER`	`SuspensionType` represents the type of suspension of a vehicle. The following semantics are assigned: — `noEntry`: No information available; — `airSuspension`: Indicating that the vehicle is equipped with an air suspension; — `hydraulicSuspension`: Indicating that the vehicle is equipped with a hydraulic suspension; — `electricSuspension`: Indicating that the vehicle is equipped with an electric suspension.

5.2.48 TariffClassId

The data type `TariffClassId` shall be of the subtype as specified in Table 48.

Table 48 — `TariffClassId`

Subtype	Parent type	Semantics
`Int4Unsigned`		`TariffClassId` provides the unique identifier of the tariff class. The default value is 0.

5.2.49 Time

The data type `Time` shall be of the subtype as specified in Table 49.

Table 49 — `Time`

Subtype	Parent type	Semantics
`Int4Unsigned`		`Time` represents the number of seconds since midnight at the start of 1970-01-01.

5.2.50 TimeClassId

The data type `TimeClassId` shall be of the subtype as specified in Table 50.

Table 50 — `TimeClassId`

Subtype	Parent type	Semantics
`Int2Unsigned`		`TimeClassId` provides the unique identifier of the time class.

5.2.51 TimeUnit

The data type `TimeUnit` shall be of the simple type as specified in Table 51.

Table 51 — `TimeUnit`

Subtype	Parent type	are
-	`INTEGER`	`TimeUnit` provides the unit of time. The following semantics are assigned: — `seconds`, — `minutes`, — `hours`, — `days`, — `months`, — `years`.

5.2.52 TrailerType

The data type `TrailerType` shall be of the simple type as specified in Table 52.

Table 52 — `TrailerType`

Subtype	Parent type	Semantics
-	`INTEGER`	`TrailerType` provides information about the type of trailer. The following semantics are assigned: — `notPresent`: Indicating trailer not attached; — `trailer`: Indicating trailer (also known as pull-bar trailer) attached; — `semiTrailer`: Indicating semitrailer (also known as articulate trailer) attached.

5.2.53 TripPurpose

The data type `TripPurpose` shall be of the simple type as specified in Table 53.

Table 53 — `TripPurpose`

Subtype	Parent type	Semantics
-	INTEGER	`TripPurpose` provides a parameter indicating the purpose of the trip of the user as pertinent to the calculation of the tariff in a contract provider specific coding. — `genericTrip`: indicating a generic trip; — `businessTrip`: indicating a trip for business purposes; — `privateTrip`: indicating a trip for private purposes; — `humanitarianAid`: indicating a trip to deliver or provide humanitarian aid; — `militaryPurposes`: indicating a trip for military purposes; — `notToBeDisclosed`: indicating that the purpose of the trip is not disclosed; — `craftBasedTrip`: indicating a trip for carrying materials, equipment or machinery for the driver's use in the course of the driver's work or used for the delivery of goods produced on a craft basis. — `combinedTransport`: indicating a trip for the transport of goods between states, or other locations as defined by the applicable law in the respective toll scheme, where the lorry, trailer, semi-trailer, with or without tractor unit, swap body or container uses the road on the initial or final leg of the journey and, on the other leg, uses rail or inland waterway or maritime services where this leg exceeds a certain distance. NOTE 1 The definition of `craftBasedTrip` is derived from the Directive (EU) 2022/362.[20] NOTE 2 The definition of `combinedTransport` is derived from the Directive 1992/106/EEC.[22]

5.2.54 TyreConfiguration

The data type `TyreConfiguration` shall be of the simple type as specified in Table 54.

Table 54 — `TyreConfiguration`

Subtype	Parent type	Semantics
-	INTEGER	`TyreConfiguration` provides information about the tyre configuration on the axles of the vehicle. The following semantics are assigned: — `notSpecified`: Indicating no values are provided; — `singleTyre`: Indicating single tyre on all axles; — `dualTyres`: Indicating dual tyres on at least one axle.

5.2.55 UserClassId

The data type `UserClassId` shall be of the subtype as specified in Table 55.

Table 55 — `UserClassId`

Subtype	Parent type	Semantics
`Int1Unsigned`		`UserClassId` provides the unique identifier of the user class.

5.2.56 VehicleAuthenticator

The data type `VehicleAuthenticator` shall be of the simple type as specified in Table 56.

Table 56 — `VehicleAuthenticator`

Subtype	Parent type	Semantics
-	`OCTET STRING`	`VehicleAuthenticator` provides an authenticator to verify that the entered vehicle data has not been altered in an unauthorized manner.

5.2.57 VehicleClass

The data type `VehicleClass` shall be of the simple type as specified in Table 57.

Table 57 — `VehicleClass`

Subtype	Parent type	Semantics
`Int1Unsigned`		`VehicleClass` provides a TSP-specific information pertaining to the vehicle. NOTE A more specific definition and an example of usage of `VehicleClass` is available in EN 15509.

5.2.58 VehicleCurrentMaxTrainWeight

The data type `VehicleCurrentMaxTrainWeight` shall be of the subtype as specified in Table 58.

Table 58 — `VehicleCurrentMaxTrainWeight`

Subtype	Parent type	Semantics
`Int2Unsigned`		`VehicleCurrentMaxTrainWeight` represents the maximum permissible weight of the complete vehicle train that is currently in operation. The value shall be stated in 10 kg units, rounded down to the next 10 kg step. NOTE 1 This weight can be lower than `VehicleTrainMaximumWeight` as it represents the current maximum train weight and not the maximum weight by design. NOTE 2 The value of `VehicleCurrentMaxTrainWeight` is calculated by adding the values of the `VehicleTechnicalPermissibleMaxLadenMass` of the tractor and potential trailer(s). If no trailers are attached, it corresponds to the `VehicleTechnicalPermissibleMaxLadenMass` of the tractor.

5.2.59 VehicleFirstAxleHeight

The data type `VehicleFirstAxleHeight` shall be of the subtype as specified in Table 59.

Table 59 — `VehicleFirstAxleHeight`

Subtype	Parent type	Semantics
`Int1Unsigned`		`VehicleFirstAxleHeight` represents the bonnet height, measured over the front axle. The value shall be stated in dm, rounded to the next dm.

5.2.60 VehicleHeightOverall

The data type `VehicleHeightOverall` shall be of the subtype as specified in Table 60.

Table 60 — `VehicleHeightOverall`

Subtype	Parent type	Semantics
`Int1Unsigned`		`VehicleHeightOverall` represents the nominal overall unladen height, in accordance with ISO 612. The value shall be stated in dm rounded to the next dm.

5.2.61 VehicleLengthOverall

The data type `VehicleLengthOverall` shall be of the subtype as specified in Table 61.

Table 61 — `VehicleLengthOverall`

Subtype	Parent type	Semantics
`Int1Unsigned`		`VehicleLengthOverall` represents the nominal maximum overall length of the vehicle in accordance with ISO 612. The value shall be stated in dm rounded to the next dm.

5.2.62 VehicleMassInRunningOrder

The data type `VehicleMassInRunningOrder` shall be of the subtype as specified in Table 62.

Table 62 — `VehicleMassInRunningOrder`

Subtype	Parent type	Semantics
`Int2Unsigned`		`VehicleMassInRunningOrder` represents the mass of the vehicle, with its fuel tank(s) filled to at least 90 % of their capacities, including the mass of the driver, fuel and liquids, fitted with the standard equipment according to the manufacturer's specifications and, when they are fitted, the mass of the bodywork, the cabin, the coupling and the spare wheels as well as the tools in accordance with ISO 1176 code ISO-M06. The value shall be stated in 10 kg units, rounded to the next 10 kg step in a way to fulfil the defined weight classes.

5.2.63 VehicleMaxLadenWeight

The data type `VehicleMaxLadenWeight` shall be of the subtype as specified in Table 63.

Table 63 — `VehicleMaxLadenWeight`

Subtype	Parent type	Semantics
`Int2Unsigned`		`VehicleMaxLadenWeight` represents the maximum permissible total weight including payload in accordance with ISO 1176 code ISO-M08. The value shall be stated in 10 kg units, rounded down to the next 10 kg step. NOTE In Europe `VehicleMaxLadenWeight` corresponds to "F.2" (maximum permissible laden mass of the vehicle in service in the Member State of registration) as defined the vehicle registration documents according to Commission Directive 1999/37/EC.[13]

5.2.64 VehicleOperationalRange

The data type `VehicleOperationalRange` shall be of the simple type as specified in Table 64.

Table 64 — `VehicleOperationalRange`

Subtype	Parent type	Semantics
`Int2Unsigned`		`VehicleOperationalRange` defines the distance a vehicle can travel under long haul transport conditions without being re-charged or re-filled. The value shall be stated in km.

5.2.65 VehicleTechnicalPermissibleMaxLadenMass

The data type `VehicleTechnicalPermissibleMaxLadenMass` shall be of the subtype as specified in Table 65.

Table 65 — `VehicleTechnicalPermissibleMaxLadenMass`

Subtype	Parent type	Semantics
`Int2Unsigned`		`VehicleTechnicalPermissibleMaxLadenMass` represents the maximum technically permissible laden mass of the tractor in accordance with ISO 1176 code ISO-M07. The value shall be stated in 10 kg units, rounded down to the next 10 kg step. NOTE In Europe `VehicleTechnicalPermissibleMaxLadenMass` corresponds to "F.1" (maximum technically permissible laden mass, except for motorcycles) as defined in the vehicle registration documents according to Commission Directive 1999/37/EC.[13]

5.2.66 VehicleTotalDistance

The data type `VehicleTotalDistance` shall be of the subtype as specified in Table 66.

Table 66 — `VehicleTotalDistance`

Subtype	Parent type	Semantics
`Int4Unsigned`		`VehicleTotalDistance` represents the total travelled distance as measured by the vehicle, in 10 m resolution, continuously incremented. NOTE The initial value of this data type can be either the value zero or the vehicle's kilometre reading at the time of personalization of the OBE.

5.2.67 VehicleTrainMaximumWeight

The data type `VehicleTrainMaximumWeight` shall be of the subtype as specified in Table 67.

Table 67 — `VehicleTrainMaximumWeight`

Subtype	Parent type	Semantics
`Int2Unsigned`		`VehicleTrainMaximumWeight` represents the maximum permissible weight of the complete vehicle train in accordance with ISO 1176 code ISO-M19. It shall be stated in 10 kg units, rounded down to the next 10 kg step. NOTE In Europe `VehicleTrainMaximumWeight` corresponds to "F.3" (maximum permissible laden mass of the whole vehicle in service in the Member State of registration) as defined in the vehicle registration documents according to Commission Directive 1999/37/EC.[13]

5.2.68 VehicleUsageCategoryType

The data type `VehicleUsageCategoryType` shall be of the simple type as specified in Table 68.

Table 68 — `VehicleUsageCategoryType`

Subtype	Parent type	Semantics
-	`INTEGER`	`VehicleUsageCategoryType` shall be used to specify the intended use of a vehicle in relation to tolling.

In addition to the data type specification the following semantic has been assigned:

— `noEntry`, indicating that there is no category available for the vehicle;

— `handicappedPeople`, indicating vehicles with special permissions for people with disabilities;

— `military`, indicating vehicles used by military forces;

— `police`, indicating vehicles used by police forces;

— `roadMaintenance`, indicating vehicles for maintaining roads and supply infrastructure;

— `circusTruck`, indicating vehicles used by travelling circus companies;

— `mobileShopTruck`, indicating vehicles used by mobile trader;

— `truckCarryingMilk`, indicating trucks for milk transportation;

— `truckCarryingTimber`, indicating timber trucks;

— `publicTransport`, indicating vehicles of local public traffic and intercity transport;

— `enforcementAgent`, indicating vehicles with special authorizations to enforce toll obligation;

— `ambulance`, indicating vehicles for medical transportation;

— `fireBrigade`, indicating vehicles used by firefighting crews;

— `officialAuthority`, indicating vehicles of public authority, government leaders or diplomatic mission;

— `agriculturalVehicle`, indicating vehicles with special characteristics for agricultural services;

— `bus`, indicating vehicles for passenger transportation;

— `mobileHome`, indicating motorhome or caravan pulled by car, van or truck;

— `mobileCrane`, indicating heavy duty vehicle with crane unit installed on it;

— `exceptionalTransport`, indicating vehicles that drive with a special traffic permission;

— `emperor`, indicating vehicles used by members of royal or imperial families;

— `vocationalVehicle`, indicating vocational vehicle;[21]

— `vehicleLiableToToll`, indicating that the vehicle is liable to toll in general ;

— `craftsTransport`, indicating that the vehicle is used for the transport of crafts;

— `combinedTransport`, indicating that the vehicle is used for combined transport.[22]

5.2.69 VehicleWeightLaden

The data type `VehicleWeightLaden` shall be of the subtype as specified in Table 69.

Table 69 — `VehicleWeightLaden`

Subtype	Parent type	Semantics
`Int2Unsigned`		`VehicleWeightLaden` represents the actual weight of the vehicle including load in 10 kg units, rounded down to the next 10 kg step.

5.2.70 VehicleWeightUnladen

The data type `VehicleWeightUnladen` shall be of the subtype as specified in Table 70.

Table 70 — `VehicleWeightUnladen`

Subtype	Parent type	Semantics
`Int2Unsigned`		`VehicleWeightUnladen` represents the nominal unladen weight in accordance with ISO 1176 code ISO-M05. The value shall be stated in 10 kg units, rounded down to the next 10 kg step.

5.2.71 VehicleWidthOverall

The data type `VehicleWidthOverall` shall be of the subtype as specified in Table 71.

Table 71 — `VehicleWidthOverall`

Subtype	Parent type	Semantics
`Int1Unsigned`		`VehicleWidthOverall` represents the nominal overall width in accordance with ISO 612. The value shall be stated in dm rounded to the next dm.

5.2.72 WeekDay

The data type `WeekDay` shall be of the simple type as specified in Table 72.

Table 72 — `WeekDay`

Subtype	Parent type	Semantics
-	`INTEGER`	`WeekDay` represents the day of the week. The following semantics are assigned: — `reserved` represents a reserved value, which shall not be used as a weekday, — `monday`, — `tuesday`, — `wednesday`, — `thursday`, — `friday`, — `saturday`, — `sunday`.

5.3 Single level data types

5.3.1 AbsolutePosition2d

The data type `AbsolutePosition2d` is an ordered list and shall consist of two components that identify a position in a 2-dimensional map as specified in Table 73.

Table 73 — `AbsolutePosition2d`

Component name	Component type	Semantics
`gnssLon`	`Longitude`, see Table 32	`gnssLon` provides the longitudinal coordinate of a geodetic position.
`gnssLat`	`Latitude`, see Table 29	`gnssLat` provides the latitudinal coordinate of a geodetic position.

5.3.2 AbsolutePosition3d

The data type `AbsolutePosition3d` is an ordered list and shall consist of two mandatory components and one optional component that identify a position in a 3-dimensional map as specified in Table 74.

Table 74 — `AbsolutePosition3d`

Component name	Component type	Semantics
`longitude`	`Longitude`, see Table 32	`longitude` provides the longitudinal coordinate of a point.
`latitude`	`Latitude`, see Table 29	`latitude` provides the latitudinal coordinate of a point.
`altitude`	`Altitude`, optional, see Table 4	`altitude` provides the altitude of a point.

5.3.3 AxleWeightLimit

The data type `AxleWeightLimit` is an ordered list and shall consist of two components that identify an axle of a vehicle and its related weight limit, as specified in Table 75.

Table 75 — `AxleWeightLimit`

Component name	Component type	Semantics
`axleNumber`	`Int1Unsigned`	`axleNumber` indicates the ordinal number of the vehicle axle, counted from the front axle.
`maxLadenWeightOnAxle`	`Int2Unsigned`	`maxLadenWeightOnAxle` represents the technically permissible maximum laden weight on the axle of the vehicle indicated in `axleNumber` in accordance with ISO 1176 code ISO-M13. The value shall be stated in 10 kg units, rounded down to the next 10 kg step. NOTE In Europe `maxLadenWeightOnAxle` corresponds to "N.1" to "N.5" (distribution of the technically permissible maximum laden mass among the axles) as defined in the vehicle registration documents according to Commission Directive 1999/37/EC.[13]

5.3.4 AxleWeightLimits

The data type `AxleWeightLimits` is an ordered list and shall consist of five components that identify the weight limits on five axles, as specified in Table 76.

Table 76 — `AxleWeightLimits`

Component name	Component type	Semantics
`maxLadenWeightOnAxle1`	`Int2Unsigned`	`maxLadenWeightOnAxle1` represents the technically permissible maximum laden weight on axle 1 of the vehicle in accordance with ISO 1176 code ISO-M13. The value shall be stated in 10 kg units, rounded down to the next 10 kg step, according to the vehicle registration documents. NOTE In Europe `maxLadenWeightOnAxle1` corresponds to "N.1" (distribution of the technically permissible maximum laden mass among the axles) as defined in the vehicle registration documents according to Commission Directive 1999/37/EC.[13]
`maxLadenWeightOnAxle2`	`Int2Unsigned`	`maxLadenWeightOnAxle2` represents the technically permissible maximum laden weight on axle 2 of the vehicle in accordance with ISO 1176 code ISO-M13. The value shall be stated in 10 kg units, rounded down to the next 10 kg step, according to the vehicle registration documents. NOTE In Europe `maxLadenWeightOnAxle2` corresponds to "N.2" (distribution of the technically permissible maximum laden mass among the axles) as defined in the vehicle registration documents according to Commission Directive 1999/37/EC.[13]
`maxLadenWeightOnAxle3`	`Int2Unsigned`	`maxLadenWeightOnAxle3` represents the technically permissible maximum laden weight on axle 3 of the vehicle in accordance with ISO 1176 code ISO-M13. The value shall be stated in 10 kg units, rounded down to the next 10 kg step, according to the vehicle registration documents. NOTE In Europe `maxLadenWeightOnAxle3` corresponds to "N.3" (distribution of the technically permissible maximum laden mass among the axles) as defined in the vehicle registration documents according to Commission Directive 1999/37/EC.[13]
`maxLadenWeightOnAxle4`	`Int2Unsigned`	`maxLadenWeightOnAxle4` represents the technically permissible maximum laden weight on axle 4 of the vehicle in accordance with ISO 1176 code ISO-M13. The value shall be stated in 10 kg units, rounded down to the next 10 kg step, according to the vehicle registration documents. NOTE In Europe `maxLadenWeightOnAxle4` corresponds to "N.4" (distribution of the technically permissible maximum laden mass among the axles) as defined in the vehicle registration documents according to Commission Directive 1999/37/EC.[13]
`maxLadenWeightOnAxle5`	`Int2Unsigned`	`maxLadenWeightOnAxle5` represents the technically permissible maximum laden weight on axle 5 of the vehicle in accordance with ISO 1176 code ISO-M13. The value shall be stated in 10 kg units, rounded down to the next 10 kg step, according to the vehicle registration documents. NOTE In Europe `maxLadenWeightOnAxle5` corresponds to "N.5" (distribution of the technically permissible maximum laden mass among the axles) as defined in the vehicle registration documents according to Commission Directive 1999/37/EC.[13]

5.3.5 DateCompact

The data type `DateCompact` is an ordered list and shall consist of three components that identify a year, a month and a day, as specified in Table 77.

Table 77 — `DateCompact`

Component name	Component type	Semantics
`year`	`INTEGER`	`year` represents the year of a date in the Gregorian calendar. The value 1990 shall only be used to provide no date.
`month`	`Month`, see Table 33	See semantic definition of data type `Month` in Table 33. The value 0 shall only be used to provide no date.
`day`	`INTEGER`	`day` represent the day of a date in the Gregorian calendar. The value 0 shall only be used to provide no date.

5.3.6 DieselEmissionValues

The data type `DieselEmissionValues` is an ordered list and shall consist of two components that identify the emission value, as specified in Table 78.

Table 78 — `DieselEmissionValues`

Component name	Component type	Semantics
`particulate`	`Particulate`, see Table 88	`particulate` represents the type and value of the particulate.
`absorptionCoeff`	`Int2Unsigned`	`absorptionCoeff` represents the corrected absorption coefficient for diesel, according to the vehicle registration documents, in 10^{-3} m^{-1}.

5.3.7 DriverCharacteristics

The data type `DriverCharacteristics` is an ordered list of two components that identify the driver's characteristics, as specified in Table 79.

Table 79 — `DriverCharacteristics`

Component name	Component type	Semantics
`driverClass`	`DriverClass`, see Table 18	See semantic definition of data type `DriverClass` in Table 18.
`tripPurpose`	`TripPurpose`, see Table 53	See semantic definition of data type `TripPurpose` in Table 53.

5.3.8 Distance

The data type `Distance` is an ordered list of two components that identify a distance, as specified in Table 80.

Table 80 — `Distance`

Component name	Component type	Semantics
`distanceValue`	`Int4Unsigned`	`distanceValue` provides the value of the distance.
`distanceUnit`	`DistanceUnit`, see Table 17	`distanceUnit` provides the unit with which the distance is measured. It defaults to kilometres.

5.3.9 Duration

The data type `Duration` is an ordered list of two components as specified in Table 81.

Table 81 — `Duration`

Component name	Component type	Semantics
`durationValue`	`Int4Unsigned`	`durationValue` provides the length of time defining a duration.
`durationUnit`	`TimeUnit`, see Table 51	`durationUnit` provides the unit of time in which the duration is measured.

5.3.10 EngineDetails,

The data type `EngineDetails` is an ordered list of two components that describe the characteristics of an engine, as specified in Table 82.

Table 82 — `EngineDetails`

Component name	Component type	Semantics
`engineCapacity`	`EngineCapacity`, see Table 20	See semantic definition of data type `EngineCapacity` in Table 20.
`enginePower`	`EnginePower`, see Table 22	See semantic definition of data type `EnginePower` in Table 22.

5.3.11 EuVehicleGroup

The data type `EuVehicleGroup` is an ordered list of two components that describe the EU vehicle group and subgroup, as specified in Table 83.

Table 83 — `EuVehicleGroup`

Component name	Component type	Semantics
`mainEuVehicleGroup`	`UTF8String`	`mainEuVehicleGroup` as defined in Commission Regulation (EU) 2017/2400,[16] Annex I.
`subGroup`	`UTF8String`	`subGroup` as defined in (EU) 2019/1242,[17] Annex I, Table 1-Vehicle sub-groups (sg).

5.3.12 ExhaustEmissionValues

The data type `ExhaustEmissionValues` is an ordered list of five components that specify emission values, as specified in Table 84.

Table 84 — `ExhaustEmissionValues`

Component name	Component type	Semantics
`unitType`	`EmissionUnit`, see Table 19	`unitType` represents the unit associated to the values in the other components.
`emissionCo`	`INTEGER`	`emissionCo` represents the exhaust emission of CO, according to the vehicle registration documents, in 10^{-3} g/km or g/kWh.
`emissionHc`	`Int2Unsigned`	`emissionHc` represents the exhaust emission of hydrocarbon (HC), according to the vehicle registration documents, in 10^{-3} g/km or g/kWh.
`emissionNox`	`Int2Unsigned`	`emissionNox` represents the exhaust emission of nitrogen oxides (NO_X), according to the vehicle registration documents, in 10^{-3} g/km or g/kWh.
`emissionHcNox`	`Int2Unsigned`	`emissionHcNox` represents the exhaust emission of HC+NO_X, according to the vehicle registration documents, in 10^{-3} g/km or g/kWh.

NOTE If the emissions are measured directly on the engine test bed, the values are declared in g/kWh.

5.3.13 FutureCharacteristics

The data type `FutureCharacteristics` is an ordered list of four components that specify vehicle parameters, as specified in Table 85.

Table 85 — `FutureCharacteristics`

Component name	Component type	Semantics
`futureElement`	`INTEGER`	`futureElement` indicates a future characteristic. No semantics is defined in this edition of the document.
`co2Scheme`	`Co2Scheme`, see Table 11	See semantic definition of data type `co2Scheme` in Table 11.
`co2Class`	`Co2EmissionClass`, see Table 8	See semantic definition of data type `co2EmissionClass` in Table 8.
`suspensionType`	`SuspensionType`, see Table 47	See semantic definition of `suspensionType` in Table 47.

NOTE The ordered list is still 8 bits long if transferred as an element encoded using the basic encoding rules. The different bits have been split up in different components according to the components defined above (fSSCCCss).

5.3.14 NumberOfAxles

The data type `NumberOfAxles` shall consist of two components that specify the number of axles in the trailer and in the tractor, as specified in Table 86.

Table 86 — `NumberOfAxles`

Component name	Component type	Semantics
`trailerAxles`	`Axles`, see Table 5	`trailerAxles` represents the number of axles of the trailer including drop axles. The value zero indicates no trailer present.
`tractorAxles`	`Axles`, see Table 5	`tractorAxles` represents the number of axles of the tractor including drop axles. A zero value indicates “not known”.

5.3.15 ObeId

The data type `ObeId` shall consist of two components that identify the manufacturer and the identifier of an OBU, as specified in Table 87. An OBE may consist of separately manufactured parts, however it can be identified, for the purpose of this document, through the identifier of the OBU.

Table 87 — `ObeId`

Component name	Component type	Semantics
`manufacturerId`	`Int2Unsigned`	`manufacturerId` provides the unique identifier of the OBU manufacturer as defined in accordance with ISO 14816. NOTE See www.itsstandards.eu/registries/register-of-manufacturers-cs2/ for a list of assigned values.
`equipmentObuId`	`EquipmentObuId`, see Table 24	Unique identifier as assigned by the manufacturer identified by component `manufacturerId`.

5.3.16 Particulate

The data type `Particulate` is an ordered list and shall consist of two components that identify the type and value of the particulate emitted by a diesel engine, as specified in Table 88.

Table 88 — `Particulate`

Component name	Component type	Semantics
`unitType`	`EmissionUnit`, see Table 19	`unitType` represents the units in which the particulate is expressed.
`value`	`INTEGER`	`value` represents the particulates for diesel, according to the vehicle registration documents.

5.3.17 PassengerCapacity

The data type `PassengerCapacity` is an ordered list of two components that indicate a vehicle's capacity in terms of passengers, as specified in Table 89.

Table 89 — `PassengerCapacity`

Component name	Component type	Semantics
`numberOfSeats`	`Int1Unsigned`	`numberOfSeats` represents the number of seats of the vehicle, including the driver's seat, according to the vehicle registration documents.
`numberOfStandingPlaces`	`Int1Unsigned`	`numberOfStandingPlaces` represents the number of standing places of the vehicle, according to the vehicle registration documents.

5.3.18 PaymentFee

The data type `PaymentFee` is an ordered list of two components that specify a payment, as specified in Table 90.

Table 90 — `PaymentFee`

Component name	Component type	Semantics
`paymentFeeAmount`	`Int2Unsigned`	`paymentFeeAmount` represents the value of the fee being charged.
`paymentFeeUnit`	`PayUnit`, see Table 35	`paymentFeeUnit` represents the unit in which the fee is expressed.

5.3.19 Period

The data type `Period` is an ordered list of two components that specify a period of time in terms of a start time and an end time, as specified in Table 91.

Table 91 — `Period`

Component name	Component type	Semantics
`beginOfPeriod`	`GeneralizedTime`	`beginOfPeriod` defines the beginning of a period.
`endOfPeriod`	`GeneralizedTime`	`endOfPeriod` defines the end of a period.

5.3.20 Provider

The data type `Provider` is an ordered list of two components that identify an organization, as specified in Table 92.

Table 92 — `Provider`

Component name	Component type	Semantics
`countryCode`	`CountryCode`, see Table 15	`countryCode` identifies the country code of the country of the national registration administrator for issuers according to ISO 14816. NOTE See www.itsstandards.eu/registries/register-of-nra-i-cs1/ for a list of assigned values.
`providerIdentifier`	`IssuerIdentifier`, see Table 28	`providerIdentifier` identifies the organization according to the national ISO 14816 register for issuers. NOTE See Reference [24] for a list of national registration administrators and their respective registers. EXAMPLE An organization can be a toll charger (TC) or a toll service provider (TSP) according to ISO 17573-1.

5.3.21 RelativePosition3d

The data type `RelativePosition3d` is an ordered list of two mandatory and one optional component that identify a position in a 3-dimensional map in relation to another position, as specified in Table 93.

Table 93 — `RelativePosition3d`

Component name	Component type	Semantics
`Longitude`	`Int2Signed`	`longitude` provides the relative longitudinal coordinate (in microdegrees) of a point with respect to a specified reference point.
`Latitude`	`Int2Signed`	`latitude` provides the relative latitudinal coordinate (in microdegrees) of a point with respect to a specified reference point.
`Altitude`	`Int2Signed`	`altitude` provides the relative altitude (in 0,25 m units) of a point with respect to a specified reference point.

5.3.22 SessionClass

The data type `SessionClass` is an ordered list of two components that identify a class and its related tariff for a vehicle in a given session, as specified in Table 94.

Table 94 — `SessionClass`

Component name	Component type	Semantics
`sessionTariffClass`	`Int1Unsigned`	`sessionTariffClass` provides a TSP-specific tariff class applied in the session.
`sessionClaimedClass`	`Int1Unsigned`	`sessionClaimedClass` provides a TSP-specific vehicle class derived from claimed characteristics.

5.3.23 SessionLocation

The data type `SessionLocation` is an ordered list of two components that identify the location of a session in terms of a direction and of a lane number, as specified in Table 95.

Table 95 — `SessionLocation`

Component name	Component type	Semantics
`ascendingKilometrage`	`BOOLEAN`	`ascendingKilometrage` provides a travel direction indicator. The following semantics are assigned: — `True`: Means "ascending kilometrage of the road". — `False`: Means "descending kilometrage of the road".
`laneCodeNumber`	`INTEGER`	`laneCodeNumber` provides a lane number.

5.3.24 SignedValue

The data type `SignedValue` is an ordered list of two components that are used to identify positive or negative values, as specified in Table 96.

Table 96 — `SignedValue`

Component name	Component type	Semantics
`positive`	`INTEGER`	`positive` represents positive integer values.
`negative`	`INTEGER`	`negative` represents negative integer values.

5.3.25 SoundLevel

The data type `SoundLevel` is an ordered list of two components that identify the levels of sound of a vehicle, as specified in Table 97.

Table 97 — `SoundLevel`

Component name	Component type	Semantics
`soundStationary`	`Int1Unsigned`	`soundStationary` represents the stationary sound of the vehicle in dB(A), according to the vehicle registration documents.
`soundDriveBy`	`Int1Unsigned`	`soundDriveBy` represents the sound of the vehicle when driving in dB(A), according to the vehicle registration documents.

5.3.26 TariffClassDescription

The data type `TariffClassDescription` is an ordered list of five optional components, the first one of which is assigned a default value, that are used to specify a tariff class, as specified in Table 98.

Table 98 — `TariffClassDescription`

Component name	Component type	Semantics
`tariffClassId`	`TariffClassId`, see Table 48	See semantic definition of data type `TariffClassId` in Table 48. If this component is not specified, its value is defaulted to 0 (zero).
`localVehicleClassId`	`LocalVehicleClassId`, see Table 30	See semantic definition of data type `LocalVehicleClassId` in Table 30.
`timeClassId`	`TimeClassId`, see Table 50	See semantic definition of data type `TimeClassId` in Table 50.
`locationClassId`	`LocationClassId`, see Table 31	See semantic definition of data type `LocationClassId` in Table 31.
`userClassId`	`UserClassId`, see Table 55	See semantic definition of data type `UserClassId` in Table 55.

5.3.27 TimeCompact

The data type `TimeCompact` is an ordered list of three components that identify a time, as specified in Table 99.

Table 99 — `TimeCompact`

Component name	Component type	Semantics
`hours`	`INTEGER`	`hours` expresses the number of hours of the time of the day after midnight.
`mins`	`INTEGER`	`mins` expresses the number of minutes after the hour of the time of the day.
`doubleSecs`	`INTEGER`	`doubleSecs` expresses the number of two-seconds after the minute of the time of the day.

5.3.28 TrailerDetails

The data type `TrailerDetails` is an ordered list of two components that specify a trailer, as specified in Table 100.

Table 100 — `TrailerDetails`

Component name	Component type	Semantics
`trailerType`	`TrailerType`, see Table 52	See semantic definition of data type `TrailerType` in Table 52.
`trailerAxles`	`Axles`, see Table 5	`trailerAxles` represents the number of axles of the trailer including drop axles. The value zero indicates no trailer present.

5.3.29 WheelsConfiguration

The data type `WheelsConfiguration` is an ordered list of four components that specify the wheels configuration of a vehicle, as specified in Table 101.

Table 101 — `WheelsConfiguration`

Component name	Component type	Semantics
`numberOfWheels`	`INTEGER`	Total number of wheels on a vehicle train. Twin-mounted tyres count as one wheel.
`numberOfDrivenWheels`	`INTEGER`	Number of driven wheels on a vehicle train.
`numberOfSteeredWheels`	`INTEGER`	Number of steered wheels on a vehicle train.
`rearWheelsSteeringType`	`RearWheelsSteeringType`, see Table 37	Type of rear steering wheels on a vehicle train.

5.4 Two-level data types

5.4.1 AxlesWeightLimits

The data type `AxlesWeightLimits` is a list of components of the same type that specify the weight limits for all axles of a vehicle, as specified in Table 102.

Table 102 — `AxlesWeightLimits`

Component name	Component type	Semantics
	List of `AxleWeightLimit`, see Table 75	List of axle weight limits.

5.4.2 ChargeObjectId

The data type `ChargeObjectId` is an ordered list of one optional component and one mandatory component that identify a charge object, as specified in Table 103.

Table 103 — `ChargeObjectId`

Component name	Component type	Semantics
`chargeObjectOperator`	`Provider`, optional, see Table 92	`chargeObjectOperator` identifies the entity operating the EFC regime in which the charge object is contained.
`chargeObjectDesignation`	`Int4Unsigned`	`chargeObjectDesignation` identifies the charge object within the given operator.

5.4.3 ContractValidity

The data type `ContractValidity` is an ordered list of two components that specify the validity of a contract, as specified in Table 104.

Table 104 — `ContractValidity`

Component name	Component type	Semantics
`contractRestrictions`	`OCTET STRING`	`contractRestrictions` contains a TSP-specific coding of the validity restrictions of a contract.
`contractExpiryDate`	`DateCompact`, see Table 77	`contractExpiryDate` contains the end-date of the validity of the contract. The validity ends at 24 h of the calendar day specified in `contractExpiryDate`. The values 1990 in the data element `year`, `noMonth` (zero) in the data element `month` and 0 (zero) in the data element `day` shall be used within the data type `DateCompact` to signal that there is no expiry date.

5.4.4 DateAndTime

The data type `DateAndTime` is an ordered list of two components that specify a date and a time, as specified in Table 105.

Table 105 — `DateAndTime`

Component name	Component type	Semantics
`timeDate`	`DateCompact`, see Table 77	`timeDate` expresses the date in accordance with the Gregorian calendar.
`timeCompact`	`TimeCompact`, see Table 99	`timeCompact` expresses time of the day in hours, minutes and seconds.

5.4.5 EnvironmentalCharacteristics

The data type `EnvironmentalCharacteristics` is an ordered list of two components that specify environmental characteristics of a vehicle, as specified in Table 106.

Table 106 — `EnvironmentalCharacteristics`

Component name	Component type	Semantics
`euroValue`	`EuroValue`, see Table 26	`euroValue` provides information about the vehicle's EURO value, according to the vehicle registration documents.
`copValue`	`CopValue`, see Table 14	`copValue` provides information about the vehicle's carbon dioxide emission values, according to the vehicle registration documents.

5.4.6 InitialVehicleRegistrationDate

The data type `InitialVehicleRegistrationDate` shall be of the parent type as specified in Table 107.

Table 107 — `InitialVehicleRegistrationDate`

Component name	Component type	Semantics
	`DateCompact`, see Table 77	`InitialVehicleRegistrationDate` expresses the date of first registration of a vehicle. NOTE In Europe `InitialVehicleRegistrationDate` corresponds to "B" (date of first registration of the vehicle) as defined in the vehicle registration documents according to Commission Directive 1999/37/EC.[13]

5.4.7 Lpn

The data type `Lpn` shall consist of three components that identify a licence plate number, as specified in Table 108.

Table 108 — `Lpn`

Component name	Component type	Semantics
`countryCode`	`CountryCode`, see Table 15	`countryCode` represents the country code associated with the licence plate number.
`alphabetIndicator`	`AlphabetIndicator`, see Table 3	See semantic definition of `AlphabetIndicator` in Table 3.
`licencePlateNumber`	`OCTET STRING`	`licencePlateNumber` contains the licence plate number coded in accordance with the alphabet indicated by the component `alphabetIndicator`.

5.4.8 PaymentMeans

The data type `PaymentMeans` is an ordered list of three components that identify the means used to perform a payment, as specified in Table 109.

Table 109 — `PaymentMeans`

Component name	Component type	Semantics
`personalAccountNumber`	`PersonalAccountNumber`, see Table 36	`personalAccountNumber` represents the number of the payment means.
`paymentMeansExpiryDate`	`DateCompact`, see Table 77	`paymentMeansExpiryDate` represents the expiry date of the payment means. Payment means expires at 24 h of `paymentMeansExpiryDate`. The values 1990 in the data element `year`, `noMonth` (zero) in the data element `month` and 0 (zero) in the data element `day` shall be used within the data type `DateCompact` to signal that there is no expiry date.
`paymentMeansUsageControl`	`OCTET STRING`	`paymentMeansUsageControl` indicates issuer's specified restrictions on the geographic usage and services allowed for the applications.

5.4.9 PaymentMeansBalance

The data type `PaymentMeansBalance` shall be of the parent type as specified in Table 110.