भारतीय मानक
Indian Standard

IS/ISO/TS 17575-3 : 2011

इलैक्ट्रॉनिक शुल्क एकत्रीकरण — स्वायत्त पद्धतियों हेतु अनुप्रयोग की अंतर्प्रष्ठ परिभाषा

भाग 3 संदर्भ आंकड़े

Electronic Fee Collection — Application Interface Definition for Autonomous Systems

Part 3 Context Data

ICS 03.220.20; 35.240.60

भारतीय मानक ब्यूरो
BUREAU OF INDIAN STANDARDS
मानक भवन, 9 बहादुरशाह ज़फर मार्ग, नई दिल्ली-110002
MANAK BHAVAN, 9 BAHADUR SHAH ZAFAR MARG
NEW DELHI-110002
www.bis.org.in www.standardsbis.in

May 2016 **Price Group 18**

Intelligent Transport Systems Sectional Committee, TED 28

NATIONAL FOREWORD

This Indian Standard (Part 3) which is identical with ISO/TS 17575-3 : 2011 'Electronic fee collection — Application interface definition for autonomous systems — Part 3: Context data' issued by the International Organization for Standardization (ISO) was adopted by the Bureau of Indian Standards on the recommendation of the Intelligent Transport Systems Sectional Committee and approval of the Transport Engineering Division Council.

This standard is published in four parts. Other parts in this series are:

Part 1 Charging

Part 2 Communication and connection to the lower layers

Part 4 Roaming

The text of ISO Standard has been approved as suitable for publication as an Indian Standard without deviations. Certain conventions and terminologies are, however, not identical to those used in Indian Standards. Attention is particularly drawn to the following:

a) Wherever the words 'International Standard' appear referring to this standard, they should be read as 'Indian Standard'.

b) Comma (,) has been used as a decimal marker while in Indian Standards, the current practice is to use a point (.) as the decimal marker.

In this adopted standard, reference appear to the following International Standard for which Indian Standard also exists. The corresponding Indian Standard, which is to be substituted in its place, is given below along with its degree of equivalence for the edition indicated:

International Standard	*Corresponding Indian Standard*	*Degree of Equivalence*
ISO 612 Road vehicles — Dimensions of motor vehicles and towed vehicles — Terms and definitions	IS 9435 : 2004 Terms and definitions relating to dimensions of road vehicles other than 2 and 3 wheelers (*first revision*)	Technically Equivalent

The technical committee has reviewed the provisions of the following International Standards/Other Publications referred in this adopted standard and has decided that they are acceptable for use in conjunction with this standard:

International Standard/ Other Publication	*Title*
ISO 1176	Road vehicles — Masses — Vocabulary and codes
ISO 4217	Codes for the representation of currencies and funds
ISO/IEC 8824-1	Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation — Part 1
ISO/IEC 8825-2	Information technology — ASN.1 encoding rules: Specification of Packed Encoding Rule (PER)
ISO/TS 12813 : 2009	Electronic fee collection — Compliance check communication for autonomous systems

International Standard/ Other Publication	*Title*
ISO 14906	Road transport and traffic telematics — Electronic fee collection — Application interface definition for dedicated short-range communication
ISO 17573	Electronic fee collection — Systems architecture for vehicle related transport services
ISO/TS 17575-1 : 2010	Electronic fee collection — Application interface definition for autonomous systems — Part 1: Charging
ISO/TS 17575-2 : 2010	Electronic fee collection — Application interface definition for autonomous systems — Part 2: Communication and connections to the lower layers
EN 15509	Road transport and traffic telematics — Electronic fee collection — Interoperability application profile for DSRC

Technical corrigendum 1 published in 2013 to the above International Standard has been given at the end of this publication.

For the purpose of deciding whether a particular requirement of this standard is complied with, the final value, observed or calculated expressing the result of a test or analysis, shall be rounded off in accordance with IS 2 : 1960 ‘Rules for rounding off numerical values (*revised*)’. The number of significant places retained in the rounded off value should be the same as that of the specified value in this standard.

Introduction

Autonomous systems

This part of ISO/TS 17575 is part of a series of specifications defining the information exchange between the Front End and the Back End in Electronic Fee Collection (EFC) based on autonomous on-board equipment (OBE). EFC systems automatically collect charging data for the use of road infrastructure including motorway tolls, zone-based fees in urban areas, tolls for special infrastructure like bridges and tunnels, distance-based charging, and parking fees.

Autonomous OBE operates without relying on dedicated road-side infrastructure by employing wide-area technologies such as Global Navigation Satellite Systems (GNSS) and Cellular Communications Networks (CN). These EFC systems are referred to by a variety of names. Besides the terms autonomous systems and GNSS/CN systems, also the terms GPS/GSM systems and wide-area charging systems are in use.

Autonomous systems use satellite positioning, often combined with additional sensor technologies such as gyroscopes, odometers and accelerometers, to localize the vehicle and to find its position on a map containing the charged geographic objects, such as charged roads or charged areas. From the charged objects, the vehicle characteristics, the time of day and other data that are relevant for describing road use, the tariff and ultimately the road usage fee are determined.

Some of the strengths of the autonomous approach to electronic fee collection are its flexibility, allowing the implementation of almost all conceivable charging principles, and its independence from local infrastructure, thereby predisposing this technology towards interoperability across charging systems and countries. Interoperability can only be achieved with clearly defined interfaces, which is the aim and justification of ISO/TS 17575.

Business architecture

This part of ISO/TS 17575 complies with the business architecture defined in ISO 17573. According to this architecture, the Toll Charger is the provider of the road infrastructure and, hence, the recipient of the road usage charges. The Toll Charger is the actor associated with the Toll Charging role. See Figure 1.

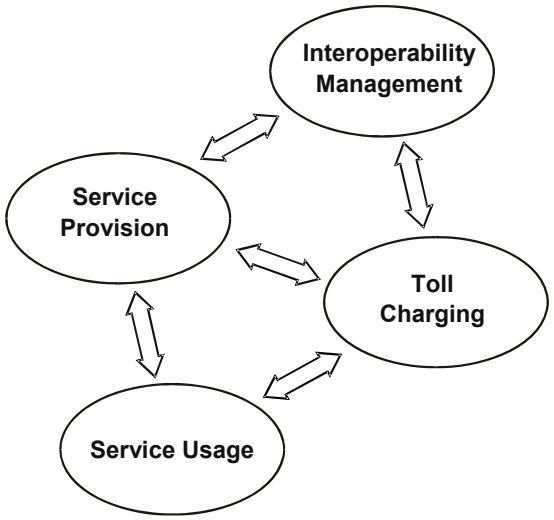


Figure 1 — The rolebased model underlying this Technical Specification

Service Providers issue OBE to the users of the road infrastructure. Service Providers are responsible for operating the OBE that will record the amount of road usage in all toll charging systems the vehicle passes through and for delivering the charging data to the individual Toll Chargers. In general, each Service Provider delivers charging data to several Toll Chargers, as well as each Toll Charger in general receives charging data from more than one Service Provider. Interoperability Management in Figure 1 comprises all specifications and activities that in common define and maintain a set of rules that govern the overall toll charging environment.

Technical architecture

The technical architecture of Figure 2 is independent of any particular practical realization. It reflects the fact that some processing functionalities can either be allocated to the OBE or to an associated off-board component (Proxy). An example of processing functionality that can be realized either on- or off-board is map-matching, where the vehicle locations in terms of measured coordinates from GNSS are associated to geographic objects on a map that either resides on- or off-board. Also tariffication can be done with OBE tariff tables and processing, or with an off-board component.

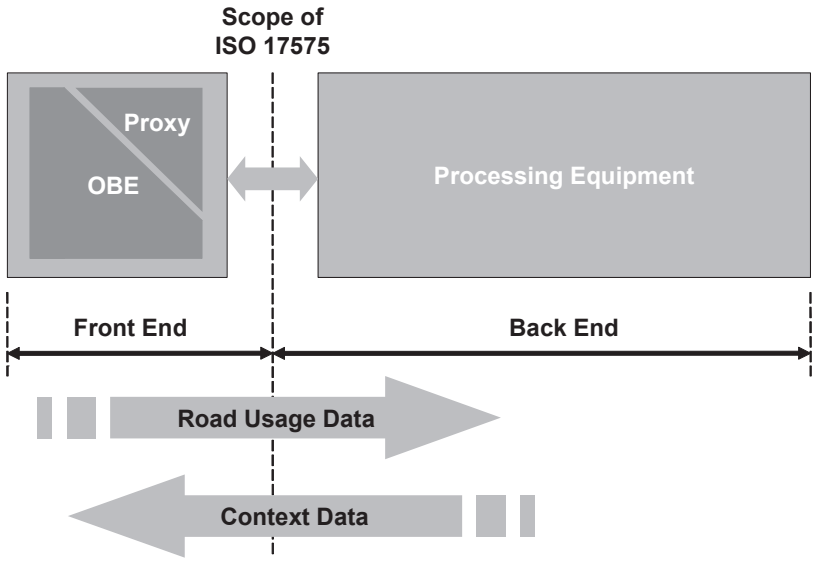


Figure 2 — Assumed technical architecture and interfaces

The combined functionality of OBE and Proxy is denoted as Front End. A Front End implementation where processing is predominately on OBE-side is known as a smart client (or intelligent client, fat client) or edge-heavy. A Front End where processing is mostly done off-board is denoted as thin-client or edge-light architecture. Many implementations between the “thin” and “thick” extremes are possible, as depicted by the gradual transition in the wedges in Figure 2. Both extremes of architectural choices have their merits and are one means where manufacturers compete with individual allocations of functionality between on-board and central resources.

Especially for thin client OBE, manufacturers might devise a wide variety of optimizations of the transfer of localization data between OBE and off-board components, where proprietary algorithms are used for data reduction and data compression. Standardization of this transfer is neither fully possible nor beneficial.

Location of the specification interface

In order to abstract from, and become independent of, these architectural implementation choices, the primary scope of ISO/TS 17575 is the data exchange between Front End and Back End (see the corresponding dotted line in Figure 2). For every toll regime, the Back End will send context data, i.e. a description of the toll regime in terms of charged objects, charging rules and, if required, the tariff scheme to the Front End, and will receive usage data from the Front End.

It has to be noted also that the distribution of tasks and responsibilities between Service Provider and Toll Charger will vary individually. Depending on local legal situation, Toll Chargers will require “thinner” or “thicker” data, and might or might not leave certain data processing tasks to Service Providers. Hence, the data definitions in ISO/TS 17575 may be useful on several interfaces.

ISO/TS 17575 also provides for basic media-independent communication services that may be used for communication between Front End and Back End, which might be line-based or an air-link, and can also be used for the air-link between OBE and central communication server.

The parts of ISO/TS 17575

Part 1: Charging, defines the attributes for the transfer of usage data from the Front End to the Back End. The required attributes will differ from one Toll Charger to another, hence, attributes for all requirements are offered, ranging from attributes for raw localization data, for map-matched geographic objects and for completely priced toll transactions.

Part 2: Communication and connection to lower layers, defines basic communication services for data transfer over the OBE air-link or between Front End and Back End.

Part 3: Context Data, defines the data to be used for a description of individual charging systems in terms of charged geographical objects and charging and reporting rules. For every Toll Charger's system, attributes as defined in Part 3 are used to transfer data to the Front End in order to instruct it which data to collect and report.

Part 4: Roaming, defines the functional details and data elements required to operate more than one EFC regime in parallel. The domains of these EFC regimes may or may not overlap. The charge rules of different overlapping EFC regimes can be linked, i.e. they may include rules that an area pricing scheme will not be charged if an overlapping toll road is used and already paid for.

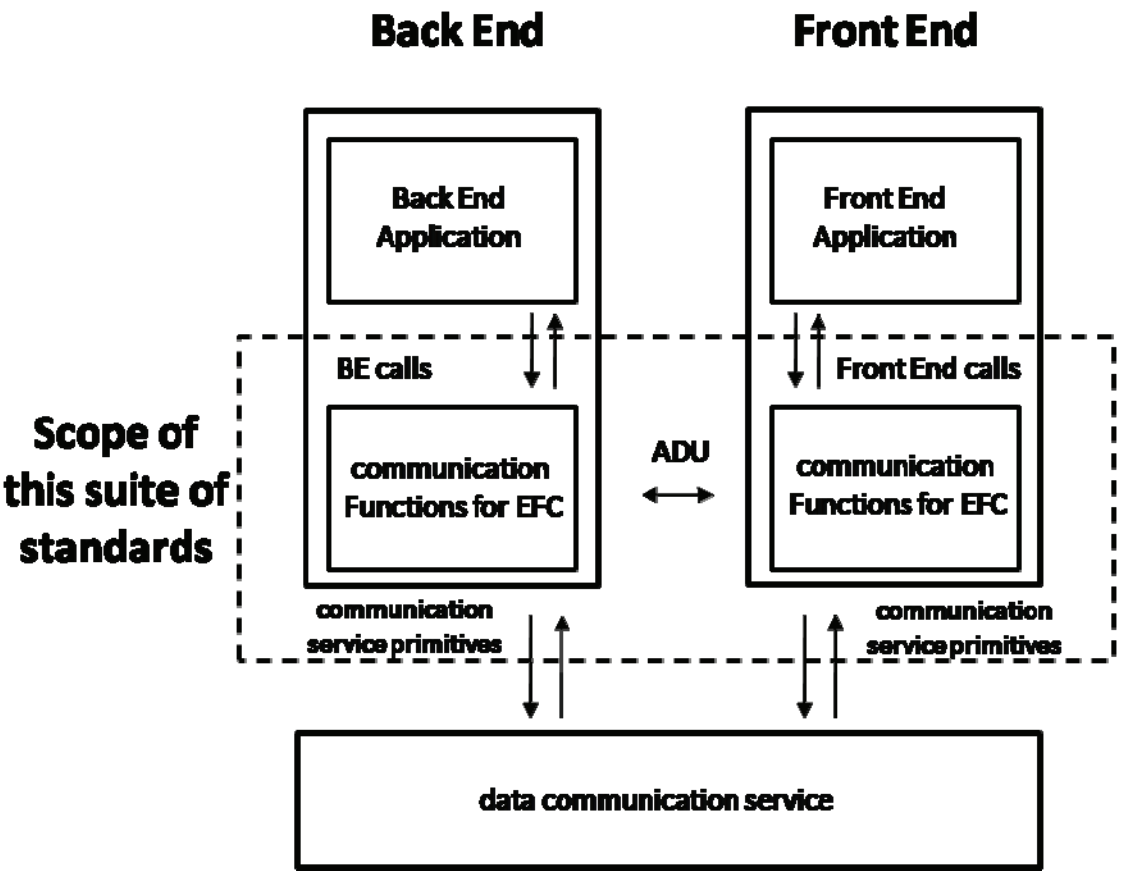


Figure 3 — Scope of ISO/TS 17575

In ISO/TS 17575, context data is the description of the properties of a single instance of an EFC context. This single instance of an EFC context operates according to one of the basic tolling principles such as

— road sectioned tolling,

— area pricing according to travelled distance,

— area pricing according to the time,

— cordon pricing.

EFC context data comprise a set of rules for charging, including the description of the charged network, the charging principles, the liable vehicles and a definition of the required contents of the charge report. This set of rules is defined individually for each EFC context according to local needs.

This part of ISO/TS 17575 contains the definitions of the above listed type of data.

Only a Front End configured with the context data necessary for the respective EFC context is able to be used for charging processes.

The following data definitions are in this part of ISO/TS 17575:

— data providing toll context overview information;

— data providing tariff information (this includes definitions of required tariff determinants like vehicle parameters, time classes and others);

— data providing context layout information;

— data providing reporting rules information.

In case one EFC domain cannot be described with a single set of context data, several of these context data are used. ISO/TS 17575-4 defines the parallel operation of more than one EFC context and how to handle interdependencies.

Applicatory needs covered by ISO/TS 17575

— The parts of ISO/TS 17575 are compliant with the architecture defined in ISO 17573.

— The parts of ISO/TS 17575 support charges for use of road sections (including bridges, tunnels, passes, etc.), passage of cordons (entry/exit), and use of infrastructure within an area (distance, time).

— The parts of ISO/TS 17575 support fee collection based on units of distance or duration, and based on occurrence of events.

— The parts of ISO/TS 17575 support modulation of fees by vehicle category, road category, time of usage, and contract type (e.g. exempt vehicles, special tariff vehicles, etc.)

— The parts of ISO/TS 17575 support limiting of fees by a defined maximum per period of usage.

— The parts of ISO/TS 17575 support fees with different legal status (e.g. public tax, private toll).

— The parts of ISO/TS 17575 support differing requirements of different Toll Chargers, especially in terms of

  — geographic domain and context descriptions,

  — contents and frequency of charge reports,

- feedback to the driver (e.g. green or red light), and
- provision of additional detailed data on request, e.g. for settling of disputes.

- The parts of ISO/TS 17575 support overlapping geographic toll domains.
- The parts of ISO/TS 17575 support adaptations to changes in
  - tolled infrastructure,
  - tariffs, and
  - participating regimes.
- The parts of ISO/TS 17575 support the provision of trust guarantees by the Service Provider to the Toll Charger for the data originated from the Front End.

Indian Standard

ELECTRONIC FEE COLLECTION — APPLICATION INTERFACE DEFINITION FOR AUTONOMOUS SYSTEMS

PART 3 CONTEXT DATA

1 Scope

This part of ISO/TS 17575 defines the content, semantic and format of the data exchange between a Front End (OBE plus optional proxy) and the corresponding Back End in autonomous toll systems. This part of ISO/TS 17575 comprises the definition of the data elements used to specify and describe the toll context details. Context data are transmitted from the Back End to the Front End.

2 Normative references

The following referenced documents are indispensable for the application 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 612, *Road vehicles — Dimensions of motor vehicles and towed vehicles — Terms and definitions*

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

ISO 4217, *Codes for the representation of currencies and funds*

ISO/IEC 8824-1, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation*

ISO/IEC 8825-2, *Information technology — ASN.1 encoding rules: Specification of Packed Encoding Rule (PER)*

ISO/TS 12813:2009, *Electronic fee collection — Compliance check communication for autonomous systems*

ISO 14906:2011, *Road transport and traffic telematics — Electronic fee collection — Application interface definition for dedicated short-range communication*

ISO 17573, *Electronic Fee Collection — Systems architecture for vehicle related transport services*

ISO/TS 17575-1:2010, *Electronic fee collection — Application interface definition for autonomous systems — Part 1: Charging*

ISO/TS 17575-2:2010, *Electronic fee collection — Application interface definition for autonomous systems — Part 2: Communication and connections to the lower layers*

EN 15509, *Road transport and traffic telematics — Electronic fee collection — Interoperability application profile for DSRC*

3 Terms and definitions

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

3.1
area pricing
charging process based on road usage occurring within a given area

3.2
attribute
application information formed by one or by a sequence of data elements, used for implementation of a transaction

NOTE Adapted from ISO 14906:2011.

3.3
authenticator
data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and/or the integrity of the data unit and protect against forgery

[ISO 14906:2011, definition 3.4]

3.4
Back End
generic name for the computing and communication facilities of the Service Provider and/or the Toll Charger

3.5
charge report
data structure transmitted from the Front End to the Back End to report road usage data and supplementary related information

3.6
charge object
any object that is part of the toll context description that may be charged for its use under certain conditions

3.7
contract
expression of an agreement between two or more parties concerning the use of the road infrastructure

[ISO 14906:2011, definition 3.7]

3.8
cordon
border line of an area

3.9
cordon pricing
charging process based on registering passages of a cordon

3.10
currencies minor unit
the minor unit of a currency (e.g. cent, pence or öre)

3.11
data element
datum, which might itself consist of lower level data elements

3.12
data integrity
property that data has not been altered or destroyed in an unauthorized manner

[ISO 7498-2:1989, definition 3.3.21]

3.13
data set
logical set of data elements selected by semantic relation

NOTE Data set is used only for better understanding and is fully independent from implementation solutions.

3.14
Front End
part(s) of the toll system where road usage data for an individual road user are collected, processed and delivered to the Back End

NOTE The Front End comprises the on-board equipment and an optional proxy.

3.15
interval scale parameters
scale of measurement of data, according to which the differences between values can be quantified in absolute but not relative terms and for which any zero is merely arbitrary

NOTE Interval scaled parameters are applicable in mathematical equations using the operators plus or minus. Interval scales having a zero offset are equal to ratio scales.

EXAMPLE The temperature scale in Celsius is an interval scale, in Kelvin it's a ratio scale.

3.16
layout
technical description of the location of a tolled object including, if applicable, auxiliary data for determining the vehicle's position relative to the tolled object

3.17
nominal scale parameters
discrete classification of data, in which data are neither measured nor ordered but subjects are merely allocated to distinct categories

NOTE Nominal scaled parameters are applicable in Boolean equations using the operators equal or not equal.

EXAMPLE A nominal scale parameter for vehicles could consist of cars, trucks, vans and motorcycles.

3.18
on-board equipment
OBE
equipment fitted within or on the outside of a vehicle and used for toll purposes

NOTE The OBE does not need to include payment means.

[ISO 14906:2011, definition 3.13]

3.19
ordinal scale parameters
scale on which data is shown simply in order of magnitude since there is no standard of measurement of differences

NOTE Ordinal scaled parameters are applicable in Boolean equations using the operators greater, greater or equal, less or less or equal.

3.20
proxy
optional component of the Front End that communicates with on-board equipment and processes road usage data into a format compliant with this part of ISO/TS 17575 and delivers the data to the Back End

3.21
ratio scale
scale of measurement of data, having a fixed zero value, which permits the comparison of differences of values

NOTE Ratio scaled parameters are applicable in mathematical equations using the operators multiplication and division.

3.22
road
any stretch of land which can be navigated by a vehicle

3.23
road usage
travelling on a road with a vehicle

3.24
road section tolling
processes for EFC based on charges for individual road sections

3.25
toll
charge, tax, fee or duty in connection with using a vehicle within a toll domain

NOTE The definition is the generalization of the classic definition of a toll as a charge, a tax, or a duty for permission to pass a barrier or to proceed along a road, over a bridge, etc. The definition above also includes fees regarded as an (administrative) obligation, e.g. a tax or a duty.

3.26
tolled area
geographic area where a toll is applied for use of vehicles

3.27
tolled passage
location where a toll is applied for passing vehicles

3.28
tolled road
road where a toll is applied for vehicles

3.29
tolled road network
road network where a toll is applied for vehicles

3.30
tolled road section
road section where a toll is applied for vehicles

3.31
toll context
logical view of a toll scheme as defined by attributes and functions

NOTE Adapted from ISO/TS 12813:2009.

3.32
toll context data
set of data necessary to define a toll context

3.33
toll domain
area or part of a road network where a toll regime is applied

[ISO 14906:2011, definition 3.21]

3.34
toll regime
set of rules, including enforcement rules, governing the collection of toll in a toll domain

3.35
toll scheme
organizational view of a toll regime, including the group of actors of one toll domain and their relationships

4 Abbreviated terms

For the purposes of this document, the following abbreviated terms apply.

ADU	Application data unit
ASN.1	Abstract Syntax Notation One (ISO/IEC 8824-1)
CCC	Compliance Check Communication, as defined by ISO/TS 12813:2009
CN	Cellular network
EFC	Electronic Fee Collection (ISO 14906:2011); here used equivalently to the term toll in ISO 17573
GNSS	Global Navigation Satellite Systems
HOT	High Occupancy Tolling
ID	Identifier
OBE	On Board Equipment
PICS	Protocol Implementation Conformance Statements
UTC	Coordinated Universal Time
VAT	Value added tax

5 General concept and overview

To enable a Front End to operate autonomously in a toll domain in the expected manner, a particular set of data elements containing application data has to be available to the Front End. These data elements shall contain a description of the rules which apply in a toll domain. This includes information regarding tariffs, vehicle classes, description of the charge objects and others.

The data elements shall be made available to the Front End using the communication services described in ISO/TS 17575-2.

For the purpose of data transfer an application data unit (ADU) is defined which comprises a header (mainly containing identification and data management information) and a data body (containing the application data elements itself).

The ADU header allows for identification of the data originator and the data sender. Furthermore it contains information about the EFC context to which the application data belong. Finally the ADU header carries a sequence number and an optional authenticator.

The data elements containing the EFC context description shall be dedicated to one single EFC context. To support the use of the Front End in multiple EFC contexts, the Front End may have the capability to manage multiple sets of data elements (one per EFC context) (see also Figure 4). Details regarding these “roaming” procedures and the normative requirements are defined in ISO/TS 17575-4.

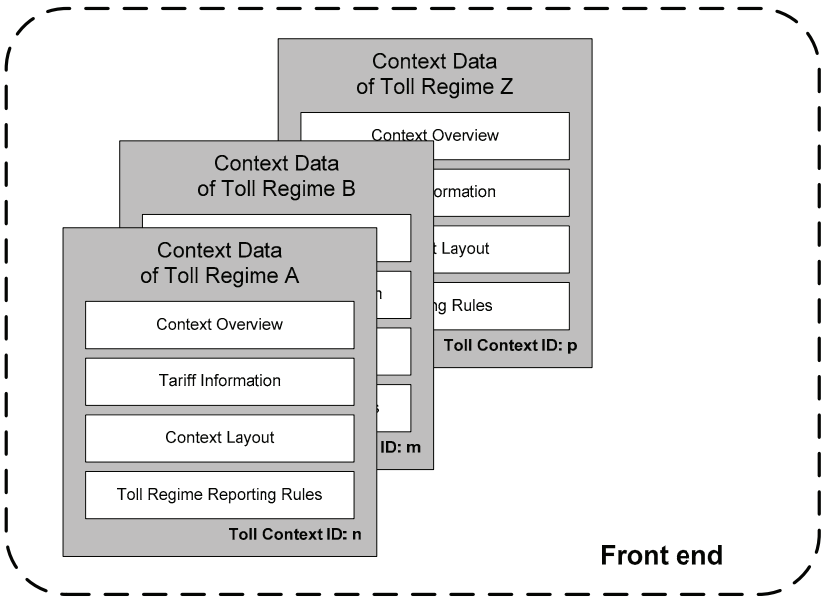


Figure 4 — Logical structure of toll context descriptions in a Front End

NOTE There may be a maximum number of toll regimes a Front End can manage. This number may depend on the memory size, the complexity of the toll regime and the envisaged use of the Front End. Front Ends may also be designed in a way to support the context description for one particular toll regime only. Other Front End designs may support context descriptions for more than one toll regime.

Context data are structured into logical data sets (refer to clause 8.2). Figure 5 gives an overview of these data sets and the type of information belonging to each data set.

Each data set comprises one or more EFC attributes. EFC attributes contain the application data. They are defined in clause 8.

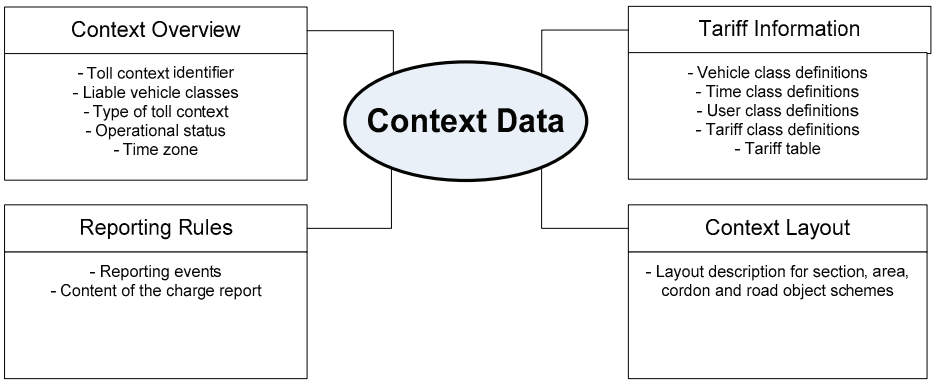


Figure 5 — Context data overview

The organization of the memory and the physical structure of the data within a Front End are outside the scope of this part of ISO/TS 17575.

6 Procedural requirements and encoding rules

6.1 Communication services

For the purpose of transmitting ADUs from the Back End to the Front End, the communication services defined in ISO/TS 17575-2 shall be used.

6.2 Version and validity handling

Each EFC attribute carries an optional data element containing version and validity information. The data type of this data element shall always be **`VersionAndValidity`**. This data type shall comprise two data elements:

- **`version`** and
- **`validFrom`**.

The data element **`version`** shall give the version number of the respective EFC attribute. The data type shall be **`VersionId`** defined in ISO/TS 17575-1. The version number shall always be used in an increasing order.

NOTE 1 This concept enables the Front End to autonomously detect missing versions of context data. This may be used to initiate an action to update the respective information.

The data element **`validFrom`** shall give the start date and time of the validity of the respective EFC attribute. The data type shall be **`DateAndTime`** as defined in ISO 14906.

The information regarding version and validity of EFC attributes enables the Front End to autonomously notice the existence of new updated context data in the Back End.

NOTE 2 Once the start date and time of a context data is reached, previous versions (having a version number lower than the current one) become obsolete. The Front End may decide - depending on local settings - to initiate an action to activate the valid context data and deactivate (and delete) the previous used version(s).

The given version and validity information are exclusively valid for the EFC attribute it belongs to.

NOTE 3 This concept allows the efficient use of different versions for different types of context data. E.g. the tariff table version may be managed independently from the one valid for context layout and reporting rules. This approach reduces the amount of data to be updated.

NOTE 4 The update process itself is outside the scope of this part of ISO/TS 17575.

6.3 Encoding rules

The data types and associated coding related to the data elements described in Clauses 7 and 8 are defined using the Abstract Syntax Notation One (ASN.1) technique according to ISO/IEC 8824-1 (Annex A).

The packed encoding rules according to ISO/IEC 8825-2 shall be applied.

Octet alignment shall be used.

The distinguished encoding rules shall not apply.

7 Application data units

7.1 Application data unit structure

For the purpose of data transfer and identification the information content shall be structured into application data units (ADU).

Each ADU shall consist of

– an ADU header and

– an ADU body.

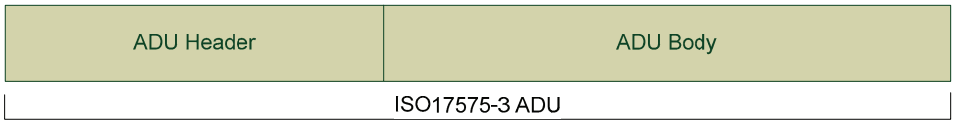


Figure 6 — Structure of an ISO 17575-3 ADU

7.2 Application data unit header

The ADU header shall consist of the data fields (Figure 7)

– information sender identifier field,

– information originator identifier field,

– toll charger identifier field

– context identifier field

– ADU sequence number field

– ADU authenticator field (optional).

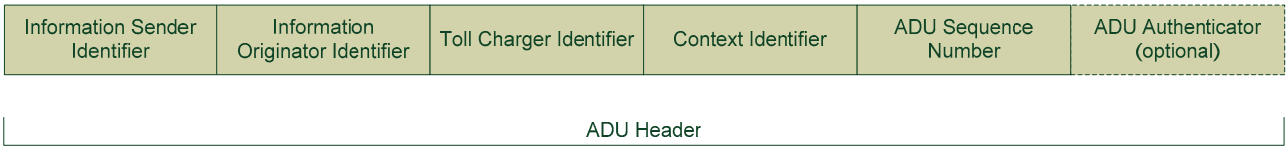


Figure 7 — Structure of the ADU header

The semantics for the data elements of the ADU header shall be according to Table 1. The data types are defined in Annex A.

Table 1 — Data elements of the ADU Header

Data element	Data type (informative)	Definition of semantic	Remarks
informationSender	Provider	Unique identifier of the entity which has sent the data provided in the ADU body	e.g. Transport Service Provider, Toll Charger or Service Provider
informationOriginator	Provider	Unique identifier of the entity which has created the data provided in the ADU body	e.g. Transport Service Provider, Toll Charger or Service Provider
tollCharger	Provider	Unique identifier of the entity which acts as toll charger of the toll regime	Consist of country code and unique number within country
contextId	ContextId	Unique identifier for the toll context the data in the ADU body are applicable to	Consist of country code and unique number within country
aduSequenceNumber	Int4	Sequence number of the respective ADU	Shall be used in increasing order. In case of overflow, the sequence number shall restart at 0.
aduAuthenticator	Bit String	Message Authenticator	Optional

7.3 Application data unit body

The ADU body shall contain one or more EFC attributes describing the EFC context. One ADU body shall contain EFC attributes belonging to one single EFC context only.

NOTE Very complex EFC contexts (e.g. containing different types of toll schemes like a cordon pricing and a section pricing) may be split into two EFC context descriptions. Interdependencies between these multiple layout descriptions are specified in ISO/TS 17575-4.

The EFC context is identified by information given in data element `contextId` in the ADU header.

EXAMPLE 1 The ADU body contains the tariff table for the Barcelona congestion charging scheme.

EXAMPLE 2 The ADU body contains the geographic description of the road network of the truck tolling schemes applicable on highways in Hungary.

EXAMPLE 3 The ADU body contains the reporting rules applicable in the nationwide all-road charging scheme in Denmark.

EFC attributes and sections descriptions are defined in chapter 8.

8 EFC Attributes

8.1 Rules with respect to support of context data

Context data available in the Front End shall contain all information required to ensure a minimum level of functionality to either participate in the services of this toll regime or to unambiguously identify the toll regime as not valid (for the specific user, vehicle, moment in time...).

Each attribute being part of the context data shall be allocated to one single EFC context.

8.2 Attributes and data sets

Each EFC context shall be described using one or more EFC attributes. EFC attributes shall contain all necessary information to enable proper functioning of the Front End in the respective EFC context.

To improve readability of this part of ISO/TS 17575 toll context description respectively attributes have been logically structured into data sets.

The following data sets are used:

- Regime Overview,
- Tariff Information,
- Context Layout,
- Reporting Rules.

NOTE Logical data sets are fully independent from the physical data structure in a Front End. The physical structure is implementation dependent and outside the scope of this part of ISO/TS 17575.

8.3 EFC attributes data catalogue

8.3.1 General

The following EFC attributes or a subset here of shall be available to the Front End (Table 2).

Table 2 — List of EFC attributes

EFC Attribute	Data set	Refer to
`TollContextOverview`	Context Overview	chapter 8.3.2
`TariffTable`	Tariff Information	chapter 8.3.3.2
`TariffClassDefinition`		chapter 8.3.3.3
`LocalVehicleClassDefinition`		chapter 8.3.3.4
`TimeClassDefinition`		chapter 8.3.3.5
`UserClassDefinition`		chapter 8.3.3.6
`TollContextLayout`	Context Layout	chapter 8.3.4
`ChargeReportingEvents`	Reporting Rules	chapter 8.3.5.1
`ChargeReportConfiguration`		chapter 8.3.5.2

In the following clauses, EFC attributes and data elements are specified in terms of:

- the names of the data elements forming the EFC attributes,
- the content and semantic definition of the EFC attributes and data elements,
- informative remarks, including references to other standards.

The specification of the corresponding data types in ASN.1 is provided in the normative Annex A.

8.3.2 Data set "Context Overview"

Toll context overview information shall be represented by one single EFC attribute **`TollContextOverview`**.

The attribute **`TollContextOverview`** shall contain information about toll context identification, type of the toll scheme (e.g. section based tolling, area pricing, cordon pricing), time zone information and information regarding the operational status of the toll context.

The main purpose of the EFC attribute **`TollContextOverview`** is to give the Front End a minimum amount of basic information regarding a toll scheme. Based on these overview data the Front End may or may not require more information.

EXAMPLE Based on the information in data element `tollContextBoundingBoxes` a Front End dedicated to a passenger car may notice that it is cruising in a toll domain. But due to the information in the data element `operationalStatus` it may realize that this toll regime is currently inactive. In this case the Front End may not require additional data (like context layout or tariff table) of this respective toll scheme.

Structure and data elements of the EFC attribute **`TollContextOverview`** are given in table below and defined in Annex A.

Table 3 — EFC attribute `TollContextOverview` (informative)

EFC Attribute	Data element	Data Type	Remark
TollContextOverview	tollCharger	TollCharger	
	tollContext	ContextId	
	tollSchemeName	UTF8String	optional
	tollSchemeType	TollSchemeType	
	operationalStatus	OperationalStatus	
	timeZone	INTEGER (-720..720)	
	tollContextBoundingBoxes	SEQUENCE OF SphericalBox	optional
	TollContextOverviewVersion	VersionAndValidity	
	TollContextOverviewAuthenticator	MessageAuthenticator	optional

The data element **`tollCharger`** shall identify the toll charger which operates the toll scheme for which the context description is valid. The data type shall be **`TollCharger`** defined in ISO/TS 17575-1:2010.

The data element **`tollContext`** shall identify the toll context for which the context description is valid. The data type shall be **`ContextId`**, defined in ISO/TS 17575-1:2010.

The data element **`tollSchemeName`** shall contain a designation for the toll scheme. The data element shall be optional. The data type shall be UTF8String.

NOTE This data element may be used to display a well known "brand name" of the toll scheme in the OBE (e.g. "LKW Maut", "Go Maut" or "TIS-PL").

The data element **`tollSchemeType`** shall contain information regarding the type of the toll scheme (road section pricing, distance based area pricing, time based area pricing, cordon pricing).

The data element **operationalStatus** shall contain information regarding the period of operation of the toll scheme. This information shall be given by defining date and time when the toll regime starts or has started operation (in data element **startsOperationAt**) and optionally by defining date and time when the toll regime will stop operation (in date element **stopsOperationAt**). Both data elements **startsOperationAt** and **stopsOperationAt** shall by of data type **DateAndTime** defined in ISO 14906:2011.

The data element **timeZone** shall give the time zone applicable for the toll scheme in relation to UTC. The data type is **INTEGER(-720..720).** The value shall be given in minutes in relation to UTC (-720 … +720 minutes). In case one single toll domain is located on more than one time zone, this toll domain shall be split into two separate toll contexts and have two context descriptions.

All absolute time information used in EFC attributes and data elements defined in this part of ISO/TS 17575 (especially in the time class definitions) shall be given in local real time. In case daylight-saving time applies the Front End shall be capable of re-calculating time information according to the applicable legal rules.

The data element **tollContextBoundingBoxes** shall contain a list of spherical rectangles which enclose the geographic area of the toll context. Each list entry shall be of data type **SphericalBox**. The data type **SphericalBox** shall contain the description of a spherical rectangle by defining the edges of the rectangle in latitude and longitude. The date element **SphericalBox** shall contain the data elements **southernLatitude**, **northernLatitude**, **westernLongitude** and **easternLongitude**. The data elements in the data type **SphericalBox** shall be used as follows:

- data element **southernLatitude** shall contain the lower (southernmost) latitude value (data type **Latitude**),
- data element **northernLatitude** shall contain the larger (northernmost) longitude value (data type **Latitude**),
- data element **westernLongitude** shall contain the lower (westernmost) latitude value (data type **Longitude**) and
- data element **easternLongitude** shall contain the larger (easternmost) latitude value (data type **Longitude**).

The data element **TollContextOverviewVersion** shall contain version and validity information for the EFC attribute **TollContextOverview**. The data type shall be **VersionAndValidity**. For details refer to clause 6.2.

The data element **TollContextOverviewAuthenticator** shall contain a message authenticator valid for the EFC attribute **TollContextOverview**. The data type shall be **MessageAuthenticator** defined in ISO/TS 17575-1:2010. The data element **TollContextOverviewAuthenticator** shall be optional.

8.3.3 Data group “Tariff Information”

8.3.3.1 Concept and overview

The EFC attributes in the data set Tariff Information shall contain all necessary information regarding

- tariff table
- tariff classes and
- locally applicable definitions for vehicle classes, time classes and user classes.

All these information is toll regime specific.

The data set shall comprise the EFC attributes

- **`TariffTable`**
- **`TariffClassDefinition`**
- **`LocalVehicleClassDefinition`**
- **`TimeClassDefinition`**
- **`UserClassDefinition`**

The following clauses give the descriptions and definitions for these EFC attributes.

The fee calculation algorithm is specified in clause 8.3.3.7.

8.3.3.2 Tariff table

Each EFC context description contains exactly one tariff table. The tariff table comprises the following information:

- For each tariff class:
    - a unique tariff class identifier,
    - a basic fee per charge unit,
    - the VAT amount per charge unit,
    - an interval scale parameter (optional),
    - an offset fee (optional)
    - a minimum fee (optional),
    - a threshold fee (optional) and
    - a maximum fee (optional)
- the applicable rounding rule (optional)
- version and validity information
- authenticator (optional).

The tariff table provides all tariff data to the Front End which are required to calculate the Fee which has to be paid by a user under given and defined conditions (vehicle class, time class, …).

8.3.3.2.1 EFC attribute TariffTable

The tariff table shall be presented in the EFC attribute **`TariffTable`**.

Structure and data elements of the EFC attribute **`TariffTable`** are given in Table 4 and defined in Annex A.

Table 4 — EFC attribute `TariffTable` (informative)

EFC Attribute	Data element	Data Type	Remark
TariffTable	tariffs	SEQUENCE OF Tariff	
	roundingRule	RoundingRule	optional
	tariffTableVersion	VersionAndValidity	
	tariffTableAuthenticator	MessageAuthenticator	optional

The data element **`tariffs`** shall contain a list of tariff related data for each tariff class. Each of the list entries shall be of data type **`Tariff`**. This data type is described in clause 8.3.3.2.2.

The optional data element **`roundingRule`** shall contain information regarding rounding of fees. The data type shall be **`RoundingRule`** which is an **`ENUMERATED`**. The following values are defined:

- **`no`** (0) - no rounding is applied,
- **`up`** (1) - always rounding up to the next larger value of the currencies minor unit,
- **`down`** (2) - always rounding down to the next lower value of the currencies minor unit,
- **`accounting`** (3) - always rounding according to methods used in accounting (e.g. defined in DIN 1333).

The data element **`tariffTableVersion`** shall contain version and validity information for the EFC attribute **`TariffTable`**. The data type shall be **`VersionAndValidity`**. For details refer to clause 6.2.

The data element **`tariffTableAuthenticator`** shall contain a message authenticator valid for the EFC attribute **`TariffTableDefinition`**. The data type shall be **`MessageAuthenticator`** defined in ISO/TS 17575-1. The data element **`tariffTableAuthenticator`** shall be optional.

8.3.3.2.2 Data type Tariff

The data type **`Tariff`** shall provide the following information per tariff class:

- a tariff class identifier (to unambiguously identify a tariff class within the toll context description),
- the charge unit (e.g. 100 m)
- monetary unit (e.g. EUR)
- basic fee per charge unit (e.g. 0,03 EUR/km),
- optional VAT amount (also given per charge unit),
- optional interval scale parameter,
- optional offset fee,
- optional minimum fee,
- optional threshold fee and
- optional maximum fee.

Structure and data elements of the data type **`Tariff`** are given in Table 5 and defined in Annex A.

Table 5 — Data type `Tariff` (informative)

Data type	Data element	Data type	Remark
Tariff	tariffClass	TariffClassId	
	chargeUnit	ChargeUnit	
	currency	Currency	
	basicFeePerChargeUnit	Real	
	vat	Int3	optional
	intervalScaleParameter	IntervalScaleParameter	optional
	offsetFee	Int3	optional
	minFee	Int3	optional
	thresholdFee	Int3	optional
	maxFee	Int3	optional

The data element **`tariffClass`** shall identify the tariff class. The data type shall be **`TariffClassId`**. This data type is defined as **`Int1`**. The applicable tariff class is defined in attribute **`TariffClassDefinition`**, which is defined in clause 8.3.3.3.

Data element **`chargeUnit`** shall contain the charge unit valid for the toll regime. It contains the smallest interval of time period, distance or number of events which is of relevance for the fee calculation. A charge unit shall have one of the following natures

- a distance (e.g. 100 m or 1 km) or
- a time period (e.g. 10 min, one hour or 7 hours) or
- a number of event occurred (e.g. one passage).

The use of charge units depend on the type of the charging scheme. The following combinations of charge unit type and toll regime type shall be possible:

- distance (e.g. meter, km) shall only be used in section or area pricing schemes,
- time (e.g. seconds, minutes, hours) shall only be used in area pricing schemes,
- event (number of passages) shall only be used in cordon or section pricing schemes

EXAMPLE 1 A section pricing scheme may use as charge unit 1 km, as the fee depends on the length of the used sections. In this example the fee increases every 1 km.

EXAMPLE 2 An area pricing scheme may use as charge unit 1 hour, as the fee depends on the duration of stay in the defined area. In this example the fee increases every hour.

EXAMPLE 3 A cordon pricing scheme may use as a charge unit 1 event, as the fee depends on the number of passages of the cordon. In this example the fee increases with every single event.

NOTE 1 In time based pricing schemes the charge unit may also be used to define the smallest possible time periods to be used in the fee calculation.

EXAMPLE 4 A time based pricing scheme with a charge unit value of 7 hours may be defined. In case the basic fee is defined by 5 GBP a stay in the charging area of less than seven hours would in any case result in a Fee of 5 GBP.

The data element **chargeUnit** shall be of data type **ChargeUnit** which is a **CHOICE** type and has the type **distance**, **time** or **event**. The data type of the data element **distance** shall be **Distance** defined in ISO/TS 17575-1. The data type of data element **time** shall be **Duration** which is defined in ISO/TS 17575-1. The data type of the data element **event** shall be **Int1**.

The data element **currency** shall identify the monetary unit applicable for this toll context and used in conjunction with the data elements **basicFeePerChargeUnit**, **vat**, **offsetFee**, **minFee**, and **thresholdFee** and **maxFee**. The data type shall be **Currency**.

The data element **basicFeePerChargeUnit** shall contain the basic fee per charge unit. The data type shall be **REAL**. The value shall be given in major units of the defined currency. The data element **basicFeePerChargeUnit** shall be given excluding VAT.

EXAMPLE 5 A **basicFeePerChargeUnit** value of 0,0175 and **currency** value of "EUR" specifies 0,0175 EUR = 1,75 Cent.

The optional data element **vat** shall contain the VAT per charge unit. The data type shall be **Int3**. The value shall be given in 0,01 %.

The optional data element **intervalScaleParameter** shall contain a list of additional parameters to adapt the basic fee depending on the parameters maximum laden weight of a vehicle and the length of the vehicle. These parameters shall contain offset values (data element **zeroOffset**), resolution data (data element **resolution**) and maximum values (data element **max**).

NOTE 2 Maximum laden weight of vehicles as interval scale parameters are currently in use in the Swiss LSVA toll scheme. The maximum length of a vehicle may be useful e.g. for payment for use of ferries.

The optional data element **offsetFee** shall be use to specify an offset value which should be added to the calculated fee (refer also to chapter 8.3.3.7). The data type shall be **Int3**. The value shall be given in 1/100 of the minor unit of the defined currency.

The optional data element **minFee** shall be use to specify a minimum fee. This minimum fee may be lower than the fee which would apply according to the real usage of the charged network. The data type shall be **Int3**. The value shall be given in 1/100 of the minor unit of the defined currency.

EXAMPLE 6 A minimum fee of e.g. 1,00 EUR may apply in cases where only a short section of an section pricing scheme has been used by a vehicle would lead to a fee of only 0,10 EUR (e.g. 1 km x 0,10 EUR/km = 0,10 EUR).

The optional data element **thresholdFee** shall be used to specify a threshold fee. In case the fee which results from the real usage of the charged network is lower than the threshold value no fee shall apply. The data type shall be **Int3**. The value shall be given in 1/100 of the minor unit of the defined currency.

EXAMPLE 7 A vehicle uses a time based area pricing scheme for 10 minutes. The fee valid for the applicable tariff class is specified by 0,02 EUR per minute. The threshold fee is specified by 0,30 EUR. As the calculated fee does not reach the threshold value the resulting fee is 0,00 EUR.

The optional data element **maxFee** shall contain a maximum value for the fee valid for the particular tariff class. The data type shall be **MaxFee**.

The data type **`MaxFee`** allows the specification of different maximum fee values

– per day,

– per week,

– per month or

– per year.

The value shall be given in minor units of the defined currency.

NOTE 3 This parameters may be used e.g. in cordon pricing schemes to cap the maximum fee due per day in case of multiple passages of the cordon border.

EXAMPLE 8 A **`maxFee`** value of 11650 and **`currency`** value of “EUR” specifies 116,50 EUR maximum fee.

8.3.3.3 Tariff class definition

Tariff classes are always considered to be toll regime specific. Tariff classes depend on one or more of the following determinants

– vehicle class

– time class

– location class

– user class.

The four tariff determinants can be defined and used completely independent from each other (orthogonal).

Each valid combination of the tariff determinants vehicle class, time class, location class and user class shall lead to one tariff class (Figure 8).

NOTE A “combination” means a logical AND of the four tariff determinants.

Different combinations of the tariff determinants may lead to the same tariff class.

EXAMPLE Both combinations {t1; v4; u34; l21} and {t4; v4;u34; l23} may lead to tariff class 345.

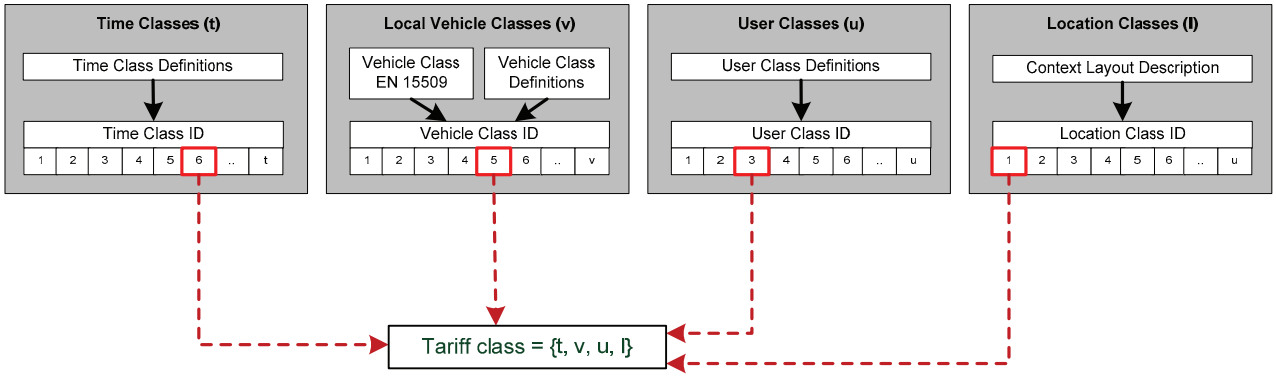


Figure 8 — Principle of creation of tariff classes

There shall be only one applicable tariff class for a user with one single contract for one single vehicle (with a given and fixed characteristic) as long as the vehicle uses one single charge object (with one single location class) at a specific point in time.

8.3.3.3.1 EFC attribute TariffClassDefinition

Tariff classes shall be defined in the EFC attribute **`TariffClassDefinition`**.

Structure and data elements of the EFC attribute **`TariffClassDefinition`** are given in Table 6 and defined in Annex A.

Table 6 — EFC attribute `TariffClassDefinition` (informative)

EFC Attribute	Data element	Data Type	Remark
TariffClassDefinition	tariffClasses	SEQUENCE OF TariffClass	
	tariffClassDefinitionVersion	VersionAndValidity	
	tariffClassDefinitionAuthenticator	MessageAuthenticator	optional

The data element **`tariffClasses`** shall contain a list of data types **`TariffClass`**. This data type is described in clause 8.3.3.3.2.

The data element **`tariffClassDefinitionVersion`** shall contain version and validity information for the EFC attribute **`TariffClassDefinition`**. The data type shall be **`VersionAndValidity`**. For details refer to clause 6.2.

The data element **`tariffClassDefinitionAuthenticator`** shall contain a message authenticator valid for the EFC attribute **`TariffClassDefinition`**. The data type shall be **`MessageAuthenticator`** defined in ISO/TS 17575-1. The data element **`tariffClassDefinitionAuthenticator`** shall be optional.

8.3.3.3.2 Data type TariffClass

The data type **`TariffClass`** shall contain all valid combinations of the tariff determinants local vehicle class, time class, location class and user class for a tariff class.

Structure and data elements of the data type **`TariffClass`** are given in Table 7 and defined in Annex A.

Table 7 — Data type `TariffClass` (informative)

Data type	Data element	Data type	Remark
TariffClass	tariffClassId	TariffClassId	
	localVehicleClasses	SEQUENCE OF LocalVehicleClassId	optional
	timeClasses	SEQUENCE OF TimeClassId	optional
	locationClasses	SEQUENCE OF LocationClassId	optional
	userClasses	SEQUENCE OF UserClassId	optional

Each tariff class shall have a unique identifier (data element **tariffClassId** of data type **TariffClassId**).

The data element **localVehicleClasses** shall contain one or more locally defined vehicle classes. The data type shall be **LocalVehicleClassId**. Vehicle classes shall be defined in EFC attribute **LocalVehicleClassDefinition** (refer to clause 8.3.3.4).

The data element **timeClasses** shall contain one or more time classes. The data type shall be **TimeClassId**. Time classes shall be defined in EFC attribute **TimeClassDefinition** (refer to clause 8.3.3.5).

The data element **locationClasses** shall contain one or more location classes. The data type shall be **LocationClassId**. Location class information shall be part of the context layout description (refer to clause 8.3.4).

The data element **userClasses** shall contain one or more user classes. The data type shall be **UserClassId**. User classes shall be defined in EFC attribute **UserClassDefinition** (refer to chapter 8.3.3.6).

EXAMPLE 1 Tariff Class 56 may be defined as the combination of Local Vehicle Class 23 and User Class 2.

EXAMPLE 2 Tariff Class 106 may be defined as the combination of Local Vehicle Classes 23, 78 and 98 Time Classes 32 and 206.

EXAMPLE 3 Tariff Class 87 may be defined as the User Class 1.

EXAMPLE 4 Tariff Class 202 may be defined as the combination of Local Vehicle Classes 156, 223 and 224 and Time Class 45 and Location Classes 2 and 5 and User Class 2.

8.3.3.4 EFC attribute LocalVehicleClassDefinition

The attribute **LocalVehicleClassDefinition** shall contain all necessary data to define vehicles classes which locally apply in the respective toll regime.

Structure and data elements of the EFC attribute are given in Table 8 and defined in Annex A.

Table 8 — EFC attribute LocalVehicleClassDefinition (informative)

EFC Attribute	Data element	Data type	Remark
LocalVehicleClass Definition	localVehicleClasses	SEQUENCE OF LocalVehicleClass	
	localVehicleClassDefinitionVersion	VersionAndValidity	
	localVehicleClassDefinitionAuthenticator	Message Authenticator	optional

The data element **localVehicleClasses** shall contain a list of data types **LocalVehicleClass**. This data type is described in clause 8.3.3.4.1.

The data element **localVehicleClassDefinitionVersion** shall contain version and validity information for the EFC attribute **LocalVehicleClassDefinition**. The data type shall be **VersionAndValidity**. For details refer to clause 6.2.

The data element **localVehicleClassDefinitionAuthenticator** shall contain a message authenticator valid for the EFC attribute **LocalVehicleClassDefinition**. The data type shall be **MessageAuthenticator** defined in ISO/TS 17575-1. The data element **localVehicleClassDefinitionAuthenticator** shall be optional.

8.3.3.4.1 Data type LocalVehicleClass

The data type **`LocalVehicleClass`** contains the definition for one vehicle class. Structure and data elements of the data type **`LocalVehicleClass`** are given in Table 9 and defined in Annex A.

Table 9 — Data type `LocalVehicleClass` (informative)

Data Type	Data element	Data Type	Remark
LocalVehicleClass	localVehicleClassId	LocalVehicleClassId	
	nominalElements	NominalVehicleParameters	
	ordinalElements	OrdinalVehicleParameters	optional
	priorityLevel	Int1	optional

Each local vehicle class shall have an unique identifier (**`localVehicleClassId`**) of the data type **`LocalVehicleClassId`**.

EXAMPLE 1 Vehicle class 21 may be defined as all trucks between 3,5 and 12 t and having 3 axles.

EXAMPLE 2 Vehicle class 107 may be defined as passenger cars having 3 or 4 axles and euro classes better than 5.

EXAMPLE 3 To exempt vehicles with very low emissions from charges e.g. in a city charging scheme, these vehicles may be grouped into one vehicle class (e.g. 67) which is defined by all vehicles with euro class 6.

In the context of tariffs vehicle specific parameters may have different nature. Vehicle specific parameters may be used as

- nominal scale parameter or
- ordinal scale parameter

The data elements **`nominalElements`** and **`ordinalElements`** (optional) shall contain the description of vehicle parameters for each local vehicle class.

Each locally defined vehicle class may depend on one or more vehicle specific parameters. Vehicle specific parameters are already defined as attributes in ISO 14906 for the purposes of EFC.

NOTE 1 This part of ISO/TS 17575 uses existing data elements (defined in ISO 14906) wherever useful.

NOTE 2 Depending on the toll regime some of the vehicle parameters may be used in a different manner, e.g. as ordinal scale parameter in one toll regime and as nominal scale parameter in a second toll regime.

The data element **`nominalElements`** shall have the data type **`NominalVehicleParameters`**. This data type is described in clause 8.3.3.4.2.

The data element **`ordinalElements`** shall have the data type **`OrdinalVehicleParameters`**. This data type is described in clause 8.3.3.4.3.

Each vehicle class may contain priority information (**`priorityValue`**) of data type **`Int1`**. This data element shall be used to determine which vehicle class is valid in case more than one vehicle class is applicable (e.g. due to overlapping class definitions). A value of 0 shall be used for the lowest priority level, a value of 255 shall be used for the highest priority level.

EXAMPLE 4 Vehicle class 45 is defined as all trucks having an overall length of 6,50 m to 12,50 m. Priority level for this class is set to 8. Vehicle class 46 is defined as all trucks having an overall length of 8,00 m to 15,00 m. Priority level for this class is set to 5. In case a vehicle would fall into both classes as it has a length of 10,65 m it has to be grouped into vehicle class 45 as the priority level of class 45 is higher.

In case vehicle classes overlap (one single vehicle falling into more than one vehicle class) the priority value shall be present. In this case the priority values of (partly) overlapping vehicle classes shall have different values.

8.3.3.4.2 Data type NominalVehicleParameters

The data type **`NominalElements`** shall contain a list of all applicable values of one or more of the following predefined data types

- **`VehicleClass`** (defined in EN 15509),
- **`VehicleAxles`** (defined in ISO 14906),
- **`EuroValue`** (defined in ISO 14906).

EXAMPLE Vehicle class 31 may be defined as all vehicles having a combination of a **`VehicleClass`** content "Trucks above 12t", a **`VehicleAxles`** content of "2 Axles" a **`VehicleAxles`** content of "3 axles" and a **`EuroClass`** content of "Euro3".

Data content, semantics and format of the data types **`VehicleClass`**, **`VehicleAxles`** and **`EuroValue`** shall be according to their definition in EN 15509 and ISO 14906, respectively.

8.3.3.4.3 Data type OrdinalVehicleParameters

The optional data type **`OrdinalVehicleParameters`** shall contain a list of the applicable value ranges of none, one or more of the following predefined optional data types

- **`VehicleLenghtOverall`** (defined in ISO 14906),
- **`VehicleHeightOverall`** (defined in ISO 14906),
- **`VehicleWidthOverall`** (defined in ISO 14906),
- **`VehicleFirstAxlesHeight`** (defined in ISO 14906),
- **`VehicleAxlesNumber`** (defined in ISO 14906),
- **`VehicleMaxLadenWeight`** (defined in ISO 14906),
- **`VehicleTrainMaximumWeight`** (defined in ISO 14906),
- **`VehicleWeightUnladen`** (defined in ISO 14906),
- **`VehicleWeightLaden`** (defined in ISO 14906),
- **`EuroValue`** (defined in ISO 14906),
- **`CopValue`** (defined in ISO 14906)
- **`VehicleClass`** (defined in EN 15509)
- **`CO2EmissionValue`**

Data content, semantics and format (including resolution if applicable) of the data types above shall be according to their definition in ISO 14906 (respectively in EN 15509 for **`VehicleClass`**).

The data element **`lowerLimit`** shall contain the lower limit of the respective parameter range. The value of the lower limit shall be included in the range.

The data element **`upperLimit`** shall contain the upper limit of the respective parameter range. The value of the upper limit shall not be included in the range.

EXAMPLE 1 A range which is defined with lowerRange = 3,5 t and upperRange = 12 t includes vehicles having 7,5 t in the respective vehicle class.

EXAMPLE 2 A range which is defined with lowerRange = 3,5 t and upperRange = 12 t includes vehicles having exactly 3,5 t in the respective vehicle class. It excludes vehicles having exactly 12 t from the respective vehicle class.

8.3.3.5 EFC attribute TimeClassDefinition

The attribute **`TimeClassDefinition`** shall contain necessary definitions for all time classes which apply in the respective toll regime.

Structure and data elements of the EFC attribute are given in Table 10 and defined in Annex A.

Table 10 — EFC attribute `TimeClassDefinition` (informative)

EFC Attribute	Data element	Data Type	Remark
TimeClassDefinition	timeClasses	SEQUENCE OF TimeClass	
	timeClassDefinitionVersion	VersionAndValidity	
	timeClassDefinitionAuthenticator	MessageAuthenticator	optional

The data element **`timeClasses`** shall contain a list of data types **`TimeClass`**. This data type is described in clause 8.3.3.5.1.

The data element **`timeClassDefinitionVersion`** shall contain version and validity information for the EFC attribute **`TimeClassDefinition`**. The data type shall be **`VersionAndValidity`**. For details refer to clause 6.2.

The data element **`timeClassDefinitionAuthenticator`** shall contain a message authenticator valid for the EFC attribute **`TimeClassDefinition`**. The data type shall be **`MessageAuthenticator`** defined in ISO/TS 17575-1. The data element **`timeClassDefinitionAuthenticator`** shall be optional.

8.3.3.5.1 Date type TimeClass

The data type **`TimeClass`** contains the definitions for one time class. Structure and data elements of the data type **`TimeClass`** are given in Table 11 and defined in Annex A.

Table 11 — Data type `TimeClass` (informative)

Data type	Data element	Data type	Remark
TimeClass	timeClassId	TimeClassId	
	nominalElements	NominalTimeParameters	optional
	ordinalElements	OrdinalTimeParameters	optional
	priorityValue	Int1	optional

Each time class shall have an identifier (**`timeClassId`**) of the data type **`TimeClassId`**.

The data elements **`nominalElements`** and **`ordinalElements`** shall contain the definitions for each time class.

All absolute time information in these data elements shall be given in local real time.

Time class definitions shall base on

- nominal scale parameter or
- ordinal scale parameter

Nominal scale parameters shall be defined in data element **`nominalElements`**. The data type is **`NominalTimeParameters`**. Nominal scale parameters shall be defined in one or more of the following data elements

- **`weekdays`**
- **`dates`**
- **`externalSetClasses`**

The data element **`weekdays`** shall be used to define one or more days of the week (e.g. Monday and Friday). Data type is **`Weekday`** which is of data type **`ENUMERATED`**.

EXAMPLE 1 All odd weekdays of the year may be coded using the nominal time parameters. This means that in data type **`NominalTimeParameters`** the element **`weekdays`** shall be present and contain the content (1), (3), (5), (7). The optional data elements **`dates`** and **`classesSetExternally`** are not present

The data element **`dates`** shall be used to identify one or more particular dates (e.g. 25.12.2017). Data type is **`DateCompact`** defined in ISO 14906.

The optional data element **`classesSetExternally`** shall be used to set any predefined time class by external input sources independently from the effective time class.

NOTE This function may be used to overrule the time class (and tariff) which would apply based on other time class determinants and the local time. This function may be used in case of a fully dynamic congestion pricing scheme. In such schemes time classes (and tariffs) may be selected based on the real time traffic situation. The external source may e.g. be the Back End or a DSRC device installed at the road side.

Ordinal scale parameters shall be defined in data element **`ordinalElements`**. The date type is **`OrdinalTimeElements`**. Ordinal scale parameters shall be defined in one or more of the following data elements

- **weekdays,**
- **absoluteTimesOfDay,**
- **relativeTimePeriods,**
- **periodsInYear**

The data element **weekdays** shall be used to define one or more periods of days during a week. The first day of the period shall be given in data element **startDay**, the last day of the period shall be given in data element **endDay**. Both data elements are of data type **Weekday** which is of data type **ENUMERATED**.

EXAMPLE 2 A specification of startDay = (1) and endDay = (5) will lead to a tariff class which is valid from Monday till Friday.

The data element **absoluteTimesOfDay** shall be used to define one or more absolute periods in time during a day (e.g. 08:00h – 10:00h and 16:30h – 18:15h). Start times shall be given in data element **startTime**, end times shall be given in data element **endTime**. Both data elements are of data type **Time** defined in ISO 14906.

The data element **relativeTimePeriods** shall be used to define one or more time periods (e.g. between 6 and 8 hours). The lower limit of the time period shall be defined in data element **minPeriod**. The upper limit of the time period shall be defined in data element **maxPeriod**. Both data elements are of data type **INT2**. The values shall be given in minutes.

The data element **periodsInYear** shall be used to define one or more periods of days within a year (e.g. 01.07. – 31.08. and 23.12. – 31.12.). The first day of the respective periods shall be given in data element **startDay**. The last day of the respective periods shall be given in data element **endDay**. Both data elements are of data type **DateCompact** defined in ISO 14906.

EXAMPLE 3 Time class 23 may be defined valid on Mondays till Friday between 08:00 h – 10:00 h and 16:00 h – 18:00 h. Priority level of this time class may be set to 18.

EXAMPLE 4 Time class 178 may be defined valid at 25.12.2015 and 26.12.2015. Priority level of this time class may be set to 245.

EXAMPLE 5 In case a toll context used time classes 23 and 178 defined in Example 1 and 2 above and the 25.12.2015 is a Tuesday, time class 178 is valid as it has the higher priority.

EXAMPLE 6 Time class 18 may be defined valid on Monday and Friday between 16:00 h and 20:15 h.

EXAMPLE 7 Time class 221 may be defined valid on Saturday between 06:00 and 21:00 and only in case the vehicle uses the toll service for less than 2 hours.

Each time class may contain an optional priority information (**priorityValue**) of data type **Int1**. This data element shall be used to determine which time class is valid in cases where more than one time class is active. A value of 0 shall be used for the lowest priority level, a value of 255 shall be used for the highest priority level.

In case time classes overlap (one specific point in time falling into more than one time class) the priority value shall be present. In this case the priority values of (partly) overlapping time classes shall have different values.

8.3.3.6 EFC attribute UserClassDefinition

The attribute **UserClassDefinition** shall contain necessary definitions for all user classes which apply in the respective toll regime.

User classes may be used to define different tariffs depending on particular characteristics of the user of toll schemes. This version of the Technical Specification supports the differentiation of users according to the contract they have and according to the number of passengers in the vehicles (to support HOT type of applications).

Structure and data elements of the EFC attribute are given in Table 12 and defined in Annex A.

Table 12 — EFC attribute `UserClassDefinition` (informative)

EFC Attribute	Data element	Data type	Remark
UserClassDefinition	userClasses	SEQUENCE OF UserClass	
	userClassDefinitionVersion	VersionAndValidity	
	userClassdefinitionAuthenticator	MessageAuthenticator	optional

The data element **`userClasses`** shall contain a list of data types **`UserClass`**. This data type is described in clause 8.3.3.6.1.

The data element **`userClassDefinitionVersion`** shall contain version and validity information for the EFC attribute **`UserClassDefinition`**. The data type shall be **`VersionAndValidity`**. For details refer to clause 6.2.

The data element **`userClassDefinitionAuthenticator`** shall contain a message authenticator valid for the EFC attribute **`UserClassDefinition`**. The data type shall be **`MessageAuthenticator`** defined in ISO/TS 17575-1. The data element **`userClassDefinitionAuthenticator`** shall be optional.

8.3.3.6.1 Data type UserClass

The data type **`UserClass`** contains the definitions for one user class. Structure and data elements of the data type **`UserClass`** are given in Table 13 and defined in Annex A.

Table 13 — Data type `UserClass` (informative)

Data type	Data element	Data type	Remark
UserClass	userClassId	UserClassId	
	contractTypes	ContractTypes	optional
	actualNumberOfPassengers	ActualNumberOfPassengers	optional

Each user class shall have an identifier (**`userClassId`**) of the data type **`UserClassId`**.

The optional data elements **`contractTypes`** and **`actualNumberOfPassengers`** shall contain the definitions for each user class.

Particular contracts which have an impact on the user class shall be defined in data element contract types. The data type shall be **`ContractTypes`**. The contracts shall be identified by the data elements **`contractProvider`** (data type **`Provider`** defined in ISO 14906) and the data element **`typeOfContract`** (data type OC**`TET STRING(SIZE(2))`**). The use of these two data types shall be according to the definitions in ISO 14906.

NOTE 1 Contract types are allocated by the Service Provider. It is up to the Service Provider to use particular contract types to support certain Toll Charger dependent user classes.

EXAMPLE 1 The use of the data element **`contractTypes`** may allow fee exceptions for fire brigades and police cars.

The actual number of passengers including the driver shall be given in the data element **actualNumberOfPassengers**. The data type shall be **ActualNumberOfPassengers**.

NOTE 2 The use of the data element **actualNumberOfPassengers** may allow for HOT type EFC applications.

User Classes shall not overlap.

8.3.3.7 The fee calculation algorithm

The fee due to pay for using a vehicle in an EFC domain is determined by the content of the element **unitsUsed** complemented by a set of four tariff determinants.

The tariff determinants are

- the vehicle class
- the time class
- the location class and
- the user class

Each of the instances of a tariff determinant is identified by an integer number. (For details how to determine these determinants see the clauses above.) At each individual moment and for a specific charge object in an EFC context a user with the current condition of his vehicle can be allocated to a single combination of these 4 tariff determinants.

The data element **TariffClass** associates a unique integer number identifier to a defined combination of the 4 tariff determinants. For each of the four tariff determinants more than one instance may be associated. These multi instances per determinant shall be combined applying an OR concatenation.

Each **tariffClass** is associated in the data element **Tariff** with a number of attributes. The relevant elements calculating the fee are **chargeUnit**, **basicFeePerChargeUnit** and **intervallScaleFactor**.

The **basicFeePerChargeUnit** is the price to be paid within a specific **tariffClass** for one single unit of the type defined in the element **ChargeUnit**. Hence, the **basicFeePerChargeUnit** defines the price e.g. for one km or 100 m or 4 hours or one specific event like crossing a border.

The **intervallScaleFactor** shall be used to scale the calculated fee linearly according to the specified parameter. Currently the **VehicleMaxLadenWeight** and the **VehicleLengthOverall** may be applied and selected using the **IntervallScaleParameter** attribute. When applying interval scale parameters it is a must that the price indicated in the **basicFeePerChargeUnit** shall be the price for a single unit of the **IntervallScaleParameter.** Hence the price per 10 kg weight or the price per decimetre length, applying the units of these attributes as defined in ISO 14906.

If the defined tariff does not use interval scaled parameters then the **intervallScaleFactor** shall be set to 1.

A data element **unitsUsed** shall be used as an accumulator for the amount of toll relevant usage of the vehicle gathered by the Front End. For the purpose of calculating the fee this accumulator must aggregate the usage only as long as the combination of all 4 tariff determinants does not change – thus the aggregated usage pertains to a single tariff class. The **unitsUsed** data element aggregates the number of “consumed” charge units. For **unitsUsed** the rounding rule as defined in data element **roundingRule** in attribute **TariffTable** shall be used, if applicable.

EXAMPLE 1 A toll scheme of the type area may use as charge unit 1 h. The charge applies for each commenced hour. In case the vehicle stays 6 hours and 23 min in this area, the units used shall be 7.

Value and unit of the charge unit are provided in the data element **`ChargeUnit`**. This is either a defined distance driven by the vehicle, either measured or read from the properties of the charge object. Or it is a defined duration a vehicle persists staying inside a toll relevant area or it is a defined number of events crossing an area border in the relevant direction.

NOTE 1 In a specific EFC context different tariff classes may be allocated to a different ChargeUnit choice. This may happen in a park house where in one tariff class, determined by a certain time class, the usage is defined and hence aggregated in hours, in another tariff class in days or in weeks. With this the more complex tariff schemes of parking areas can be modelled.

This completes the relevant elements required to calculate the fee according to a single tariff class. The total fee for a complete charge report interval shall be calculated aggregating all the fees calculated for a single tariff class belonging to the full charge report period.

For the single tariff fee calculation the following algorithm shall be applied:

$$Fee_{[period\ n]} = unitsUsed_{[period\ n]} \cdot basicFeePerChargeUnit_{[TariffClass\ n]} \cdot intervalScaleFactor$$

whereas

- *Fee* is the resulting fee (excluding VAT) for a single tariff class in the defined currency,
- *unitsUsed* contains the integer number of detected/measured/consumed charge units (the relevant charge unit is given in date element **`chargeUnit`** comprising its value and unit; e.g. 100 m or 2 hours or 1 mile),
- *basicFeePerChargeUnit* is the fee (excluding VAT) which applies for the use of the defined charge unit given in major units of the defined currency. It is given in data element **`basicFeePerChargeUnit`**. The applicable monetary unit is defined in data element **`currency`**. (e.g. **`basicFeePerChargeUnit`** = 0,1223 and **`currency`** = SEK defines a basic fee per defined charge unit of 0,1223 SEK).
- *intervalScaleFactor* is the value of the selected interval scale parameter using the information given in the data element **`intervalScaleParameter`**
- *TariffClass* is the identifier associated to the combination of the applicable TimeClass, LocalVehicleClass, UserClass and LocationClass which is defined in the data type **`tariffClass`**

NOTE 2 The above formula does not include optional **`offsetFee`**, **`minFee`**, **`thresholdFee`** and **`maxFee`**.

EXAMPLE 2 In a distance based pricing scheme a vehicle with the tariff class 25 uses the toll road network for 3,4 km. The tariff for tariff class 25 is defined by 0,159 EUR/km. The fee shall be determined in 100 m resolution. Therefore the charge unit is set to 100 m. The **`basicFeePerChargeUnit`** will be 0,0159 and **`currency`** will be EUR (0,159 EUR/km = 0,0159 EUR/100 m). The used units is 34 (= 3,4 km/100 m). Hence the fee results in 34 x 0,0159 EUR = 0,5406 EUR.

EXAMPLE 3 In an area pricing scheme a vehicle with the tariff class 9 uses the tolled area for 8 hours. The tariff for the tariff class 9 is defined by 2,99 GBP/2 hours. The fee shall be determined in 2 hours resolution. Therefore the charge unit is set to 2 hours. The **`basicFeePerChargeUnit`** will be 2,99 and **`currency`** will be GBP. The used units is 4 (= 8 hours/2 hours). Hence the fee results in 4 x 2,99 GBP = 11,96 GBP.

The actual value used for the **`intervalScaleFactor`** shall be the value applicable during the calculation period for either the parameter **`VehicleMaxLadenWeight`** respectively the value of the parameter **`VehicleLengthOverall`** plus the value indicated by the parameter **`zeroOffset`** and rounded to the resolution indicated by the parameter **`resolution`**. The value used shall also be capped to the value indicated by the parameter **`max`**.

Additional flat rates like monthly fees shall be applied after calculating the fee according to these rules. This part of ISO/TS 17575 does not support these processes.

8.3.4 Data set “Context Layout”

Toll context layout information shall be provided by one single EFC attribute **`TollContextLayout`**.

The attribute **`TollContextLayout`** shall contain all necessary descriptions of the tolled road infrastructure of an EFC context. It shall contain a list of single charge objects including their identifiers, their geographic description and additional information like applicable location classes.

The main purpose of the EFC attribute **`TollLayoutContext`** is to give the Front End all information which enables it to detect (and measure) the usage of a tolled infrastructure.

Depending on the type of the toll regime the attribute **`TollContextLayout`** shall contain the layout descriptions of one of the following types:

- section pricing scheme
- area pricing scheme
- cordon pricing scheme.

NOTE Pricing for road objects (e.g. tunnel, ferries, passes) are considered as subtypes of section pricing schemes. For the description of road objects means provided for the description of sections shall be used.

Structure and data elements of the EFC attribute are given in Table 14 and defined in Annex A.

Table 14 — EFC attribute `TollContextLayout` (informative)

EFC Attribute	Data element	Data Type	Remark
TollContextLayout	layoutDescription	Layout	
	geoRefPoint	Point	optional
	tollContextLayoutVersion	VersionAndValidity	
	tollContextLayoutAuthenticator	MessageAuthenticator	optional

The data element **`layoutDescription`** shall give the description of the layout of the respective EFC context. The data type shall be **`Layout`**.

The data type **`Layout`** contains one of the data elements **`sectionLayout`** (a list of data types **`SectionLayout)`**, **`areaLayout`** (a list of data types **`AreaLayout`**) and **`cordonLayout`** (a list of data types **`CordonLayout`**).

These data elements shall be present according to the type of the toll context following the rules below:

- For section or road object pricing schemes the data element **`sectionLayout`** shall be used. The data element is described and defined in clause 8.3.4.1.
- For area pricing schemes the data element **`areaLayout`** shall be used. The data element is described and defined in clause 8.3.4.2.
- For cordon pricing schemes the data element **`cordonLayout`** shall be used. The data element is described and defined in clause 8.3.4.3.

The optional data element **geoRefPoint** (data type **Point**) may be used as geographic reference point for the whole EFC context. This reference point shall be specified by its longitude and latitude value. In case this data element is present all longitude and latitude values given in the layout description shall be relative to this point.

The data element **tollContextLayoutVersion** shall contain version and validity information for the EFC attribute **TollContextLayout**. The data type shall be **VersionAndValidity**. For details refer to clause 6.2.

The data element **tollContextAuthenticator** shall contain a message authenticator valid for the EFC attribute **TollContextLayout**. The data type shall be **MessageAuthenticator** defined in ISO/TS 17575-1. The data element **tollContextAuthenticator** shall be optional.

8.3.4.1 Data element sectionLayout/Data type SectionLayout

The data element **sectionLayout** in the data type **Layout** contains a list of data types **SectionLayout**. This list shall contain the geographic description and supporting data of all road sections forming the charged road network of the respective toll regime. Each list entry represents one road section. The entire charged road network may consist of one or more than one sections.

EXAMPLE 1 Austrian Go-Maut and German LKW-Maut systems are considered as section tolling schemes. Both tolling schemes consist of a certain number of road sections. In both EFC schemes the fee depends on predefined length information per section.

A tolled road section shall be described by the following data

- Section identifier/charge object identifier,
- Section/charge object name (optional),
- Geographic reference point (optional),
- List of points used in the data elements of the section description,
- Description of the toll path,
- Liability rules,
- List of possible paths towards the toll path (optional),
- List of possible paths onwards the toll path (optional),
- Supporting information regarding other road network located close to the toll section (optional),
- Charged distance of the section (may not in all cases equal the real distance),
- Real distance of the section (optional),
- Time classes applicable for this section (optional) and
- Location class for the section.

Each section shall be described by means of the data type **SectionLayout.**

Structure and data elements of the data type **SectionLayout** is given in Table 15 and defined in Annex A.

Table 15 — Data type `SectionLayout` (informative)

Data type	Data element	Data type	Remark
SectionLayout	chargeObjectId	ChargeObjectId	
	chargeObjectName	UTF8String	optional
	chargeObjectRefPoint	Point	optional
	networkPoints	SEQUENCE OF PointIdDefinition	optional
	tollPath	Link	
	liabilityRules	LiabilityRules	
	pathStructureTowards	SEQUENCE OF Link	optional
	pathStructureOnwards	SEQUENCE OF Link	optional
	supportingInformation	SEQUENCE OF SupportingPoint	optional
	chargeDistance	ChargeDistance	
	realDistance	Distance	optional
	applicableTimeClasses	SEQUENCE OF TimeClassId	optional
	locationClass	LocationClassId	
	storageRequired	BOOLEAN	optional

Each section shall be identified by a charge object identifier. The data element **`chargeObjectId`** shall contain this identifier. The data type shall be **`ChargeObjectId`**. The charge object identifier shall be unique within the toll context (within the given **`tollContextId`**). In case the section being part of a road network which is defined within an area pricing scheme the charge object identifier shall be unique within the area (within the given **`areaId`**).

The optional data element **`chargeObjectName`** shall be used to give the name or designation of the section. The data type shall be **`UTF8String`**.

EXAMPLE 2 “A8 Augsburg West – Augsburg Ost” or “M9 Exits 4a – 5”

The optional data element **`chargeObjectRefPoint`** shall be used to set a reference point valid for this particular charge object. In case this data element is present all longitude and latitude values given within this charge object description are considered being relative to this point.

NOTE 1 To set a reference point valid for the entire context layout the data element **`geoRefPoint`** in the EFC attribute **`TollContextLayout`** shall be used.

The optional data element **`networkPoints`** shall contain a list of all geographic points which are used for the description of the section. Each list element shall be of the data type **`PointIdDefinition`**. For each point this list shall contain a unique identifier and the respective longitude and latitude values.

NOTE 2 In case the list of network points is present and the points are given by their coordinates and identifiers, other data elements in this attribute may reference single points by their identifiers only.

The data element **tollPath** shall contain the geographic description of the respective road section. The data type shall be **Link**. This data type is defined in clause 8.3.4.1.1.

The data element **liabilityRules** shall contain rules to specify under which circumstances a vehicle using the section is liable to pay for the use of the entire toll section. The data type shall be **LiabilityRules**. This data type is defined in clause 8.3.4.1.2.

The optional data element **pathStructureTowards** shall contain the layout description of the roads which are in front (relative to the driving direction) of the charged road section. It shall contain a list of data types **Link**. For the definition of the data type **Link** refer to clause 8.3.4.1.1.

NOTE 3 This information may be useful for support certain charge object detection algorithms in some Front End implementations.

The optional data element **pathStructureOnwards** shall contain the layout description of the roads which are connected ahead (relative to the driving direction) of the charged road section. It shall contain a list of data types **Link**. For the definition of the data type **Link** refer to clause 8.3.4.1.1.

NOTE 4 This information may be useful for support certain charge object detection algorithms in some Front End implementations.

The optional data element **supportingInformation** shall contain a list of intermediate points on the charged road section. For each point in the list a distance to other (non-tolled) road infrastructure may be given. The data type of the list elements is **SupportingPoint**. This data type is described in clause 8.3.4.1.3.

The data element **chargeDistance** shall contain length information for the toll section. This length information shall be used as the basis for calculation of the fee to be paid for the use of this road section. However, this length information may or may not represent the real length of the section. The type of the data element **chargeDistance** shall be **ChargeDistance.**

The definition of the data type **ChargeDistance** allows either a predefined distance value for the section or a distance values depending on the entry section of the road network.

NOTE 5 A predefined length value may for instance be set due to “political” reasons or due to rounding rules.

NOTE 6 Section length values which depend on the entry section may be used in closed tolling schemes.

The optional data element **realDistance** shall give the real measured length of the toll section. This length information may be different from the one given in data element **chargeDistance**. The type of the data element **realDistance** shall be **Distance** defined in ISO/TS 17575-1.

The optional data element **applicableTimeClasses** shall contain a list of all time classes (represented by data type **TimeClassId**) which may apply to this specific toll section.

NOTE 7 This data element enables single road sections being charged at certain times only.

EXAMPLE 3 This supports highway sections in urban areas which will be charged only at peak hours.

The data element **locationClass** shall contain the location class information for this toll section (data type **LocationClassId**). The value in this data element shall be used as one of the tariff class determinants (refer to chapter 8.3.3.3).

8.3.4.1.1 Data type Link

The data type **Link** shall contain the description of a certain piece of road. Each link shall have an unique identifier. Each link shall be described by a start point and an end point.

To better reflect to reality (curves etc.) a link may contain one or more intermediate points (Figure 9).

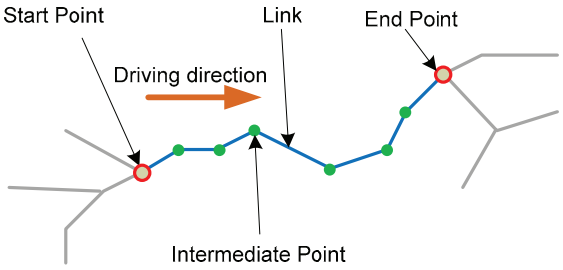


Figure 9 — Elements of a link

The data type **`Link`** shall contain the data elements **`linkId`**, **`startPoint`**, **`endPoint`** and **`intermediatePoints`**.

The data element **`linkId`** shall contain the identifier of the link. It shall be unique within the one toll context. The data type shall be **`LinkId`**.

The data element **`startPoint`** shall contain the definition of the start point of the link. The data element **`endPoint`** shall contain the definition of the end point of the link. Both data elements shall be specified by either using their point identifiers (data element **`pointIdentifier`**) or by their geographic coordinates using absolute coordinates (data element **`absolutePointCoordinates`**) or coordinates relative to the defined reference point (**`relativePointCoordinates`**).

NOTE In case relative coordinates are chosen positions can only be specified in an area of ± 3,5 km.

A link shall always be one-directional. The direction is implicitly given by start point and end point.

The optional data element **`intermediatePoints`** shall contain a list of one or more than one intermediate points which are on the piece of road and may be used to better model the real run of the piece of the road. Each point in the list shall be specified by either using their point identifiers (data element **`pointIdentifier`**) or by their geographic coordinates (data element **`absolutePointCoordinates`** or **`relativePointCoordinates`**).

In case the data type **`Link`** is used to describe a more complex road structure in the data elements **`pathStructureTowards`** and **`pathStructureOnwards`** the following rule shall apply.

Links which physically connect to each other shall be identified by using the same point as end point of the first link and the start point of the second link (refer to Figure 10). Depending on the referencing method chosen in the data type **`Link`** this shall be done by one of the following methods:

– Same longitude and latitude values in data elements **`absolutePointCoordinates`** and **`relativePointCoordinates`** for both the end point of the first link and the start point of the connected link

– Same point identifier in the data element **`pointIdentifier`** for both the end point of the first link and the start point of the connected link

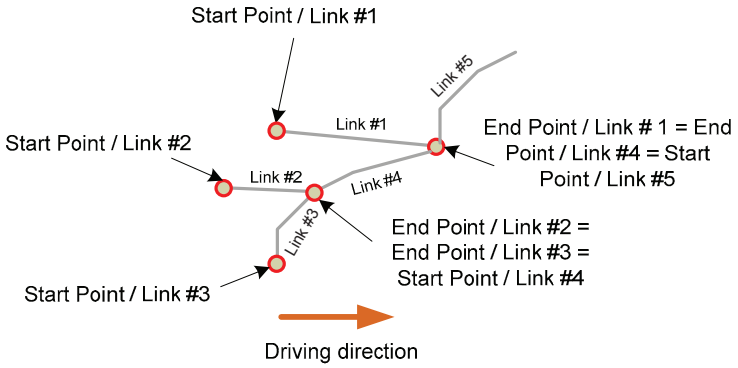


Figure 10 — Example of a complex road structure

8.3.4.1.2 Data type LiabilityRules

To define the liability rule which applies in the tolling context the data type **`LiabilityRules`** shall be used. Liability rules give the conditions to decide if and when a vehicle driving on the section has to pay for the section or not. This version of this part of ISO/TS 17575 supports the following rules:

– passage of defined number (one or two) of location points (toll point(s)) on the entire section,

– passage of a particular road piece of the entire section,

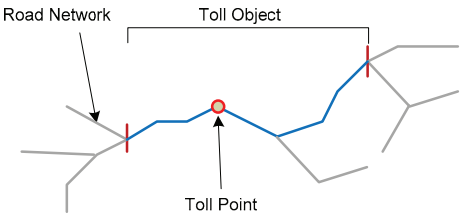
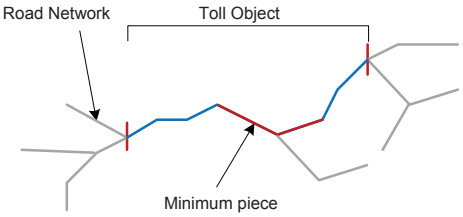
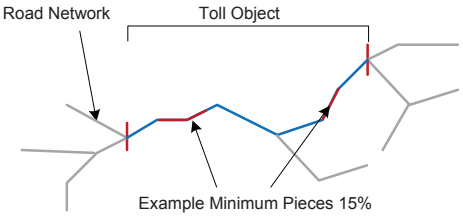
– passage of a certain percentage of the entire section

NOTE Liability rules are typically defined in a legal framework belonging to each toll scheme. This part of ISO/TS 17575 only gives the possibility to “convert” the legal rules into technical definitions.

The data type **`LiabilityRules`** shall contain one of the data elements **`tollPoints`**, **`minTollPath`** and **`minTollPortion`**.

The table below contains a description of the use of the data type **`LiabilityRules`**.

Table 16 — Use of the data elements in data type `LiabilityRules`

Rule	Description	Example	ASN.1 element in `liabilityRules`
Toll Point(s)	Passage of a number of location points (one to two) makes the vehicle liable to pay for the use of the entire section		`tollPoints`
Absolute piece of section	Passage of a certain piece of the entire toll section makes the vehicle liable to pay for the use of the entire section		`minTollPath`
Relative piece of section	Passage of a certain percentage of the entire toll section makes the vehicle liable to pay for the use of the entire section		`minTollPortion`

The optional data element `tollPoints` shall be used to define toll point(s). It shall contain one or two points. Both points shall be specified by either using their point identifiers (data element `pointIdentifier`) or by their geographic coordinates (data element `absolutePointCoordinates` and `relativePointCoordinates`.

The optional data element `minTollPath` shall be used to define a particular piece of the section which shall be passed as a minimum. It shall contain the geographic description of this piece of road. The data type shall be `Link`.

The optional data element `minTollPortion` shall be used to define the portion of the entire section (percentage value) which has to be passed as a minimum. The data type shall be `INTEGER(0..1000)`. The resolution shall be 1 per mille (1 ‰).

8.3.4.1.3 Data type SupportingPoints

The data type `SupportingPoints` shall be used to give information regarding the distance to road infrastructure which is adjacent to the charge object. It shall contain a list of points which shall be located on the charged section. Each point shall be specified by either using their point identifiers (data element `pointIdentifier`) or by their geographic coordinates (data element `absolutePointCoordinates` or `relativePointCoordinates)`.

For each point the minimum distance to the adjacent road infrastructure shall be given in the optional data element `distanceToNextRoad`. This data element shall be of type `Distance` defined in ISO/TS 17575-1.

8.3.4.2 Data element areaLayout

The data element **areaLayout** shall give the full description of the context layout of an area tolling scheme. The data element **areaLayout** shall contain a list of the descriptions of one or more geographic area forming one toll context and belonging to one toll regime (applying one single tariff table, tariff class definitions,...). In case more the one area layout is present the areas must not overlap. The elements of the list shall have the type **AreaLayout**.

EXAMPLE An area tolling scheme may be split into two non overlapping areas which may no direct connection to each other. One may e.g. cover the Southern part of a metropolitan area, the second one may cover the Northern part, whereas the city centre is not covered at all.

Each area shall be described using the data type **AreaLayout**.

Structure and data elements of the data type **AreaLayout** is given in Table 17 and defined in Annex A.

Table 17 — Data type AreaLayout (informative)

Data type	Data element	Data Type	Remark
AreaLayout	areaId	AreaId	
	areaBorder	Polygon	
	locationClass	LocationClassId	
	roadNetwork	SEQUENCE OF RoadNetwork	optional

The data element **areaId** shall contain an identifier for the area. The identifier shall be unique in the toll context. The data type shall be **AreaId**.

The data element **areaBorder** shall contain the description of the border of the area. The data type shall be **Polygon**.

The data type **Polygon** is defined as a list of the data type **Point**. The points shall be defined in an order which creates and closes the polygon in a clockwise direction. The segments of the polygon shall be created by connecting each of the points with its successor point. The polygon shall be closed by connecting the last point in the list with the first point. The points shall be defined in a way that the connections between the points don't intersect. The inner side of the polygon (tolled area) is defined as the geographical area which is located at the right hand side when moving on a polygon segment from its start point to its end point.

The data element **locationClass** shall contain the location class for this area (data type **LocationClassId**). The value in this data element shall be used as one of the tariff class determinants (refer to chapter 8.3.3.3).

The optional data element **roadNetworks** shall contain a list of descriptions of road networks. Each of these networks may have a different location class. This data element may be used for area pricing schemes which charge according to the measured distance on a specified road network. Moreover the area may contain more than one network supporting different tariffs.

NOTE This feature may be used to enable different tariffs e.g. depending on the classes of the road networks. In such a case all road segments belonging to the same road class shall be defined in one single data element **roadNetworks** which has a defined **locationClass**.

Each list entry shall have the data type **RoadNetwork**. The data elements and respective data types of the data type **RoadNetwork** are described in clause 8.3.4.1.

The charge object identifiers allocated to each section of the road network shall be unique within the entire EFC context.

8.3.4.3 Data element cordonLayout

The data element **`cordonLayout`** shall give the full description of the context layout of a cordon tolling scheme. The data element **`cordonLayout`** shall contain a list of the descriptions of one or more geographic cordons forming one toll context and belonging to one toll regime (applying one single tariff table, tariff class definitions,...). In case more the one cordon layout is present the cordons must not overlap. The elements of the list shall have the type **`CordonLayout`**.

Each cordon shall be described using the data type **`CordonLayout`**.

Structure and data elements of the data type **`CordonLayout`** is given in Table 18 and defined in Annex A.

Table 18 — Data type `CordonLayout` (informative)

Data type	Data element	Data Type	Remark
CordonLayout	cordonId	CordonId	
	cordonBorderPolygon	SEQUENCE OF CordonBorderSegment	

The data element **`cordonId`** shall contain an identifier for the cordon. The identifier shall be unique in the toll context. The data type shall be **`CordonId`**.

The data element **`cordonBorderPolygon`** shall be used for the description of the border of the cordon. It is defined in chapter 8.3.4.3.1.

8.3.4.3.1 Data type CordonBorderPolygon

The data type **`CordonBorderPolygon`** shall give the full description of border of tolled cordon. It is defined as a **`SEQUENCE OF CordonBorderSegment`** containing the segments of the cordon border polygon.

The segments of the cordon border shall be defined in clockwise order. Each segment shall be defined by

- a segment identifier
- a starting point
- an optional element for giving entry point details
- an optional element for giving exit point details

The segments of the cordon border polygon shall be created by connecting the start point of the segment with the start point of the succeeding segment. The segments shall be defined in an order which creates and closes the polygon in a clockwise direction. The polygon shall be closed by connecting the start point of the last segment with the start point of the first segment. The points shall be defined in a way that the segments don't intersect each other. (Figure 11)

The inner side of the cordon border polygon is defined as the geographical area which is located at the right hand side when moving on a polygon segment from its start point to its end point.

At locations where cordon segments and road networks intersect, an entry and/or exit point is created. A cordon segment shall be designed in a manner that either one entry point or one exit point or one combined entry/exit point is created. Cordon segments shall not carry more than one entry point or one exit point or one combined entry/exit point (refer to Figure 11).

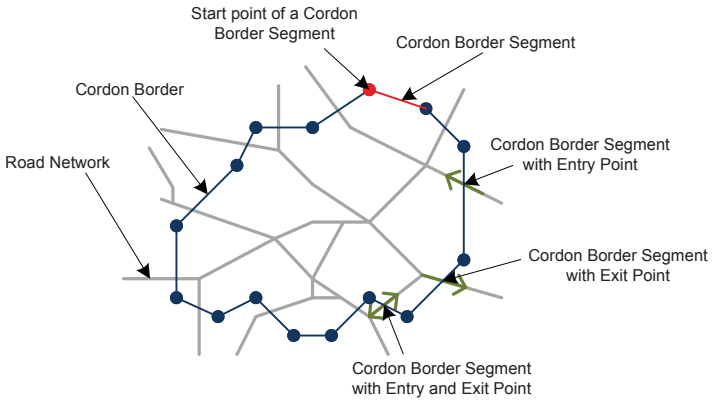


Figure 11 — Elements of a cordon layout description

Structure and data elements of the data type `CordonBorderSegment` is given in Table 19 and defined in Annex A.

Table 19 — Data type `CordonBorderSegment` (informative)

Data type	Data element	Data Type	Remark
CordonBorderSegment	cordonSegmentId	CordonSegmentId	
	startPoint	Point	
	cordonEntryData	SEQUENCE	optional
	cordonExitData	SEQUENCE	optional

The data element `cordonSegmentId` shall contain an identifier for the cordon segment. The identifier shall be unique within the cordon layout. The data type shall be `CordonSegmentId`.

The data element `startPoint` shall be used for the description of the start point of the border segment. The data type shall be `Point`.

The optional data element `cordonEntryData` shall be present in case the respective cordon border segment is considered as an entry location of the tolled cordon.

The data element `entryLocationId` shall contain a charge object identifier. The data type shall be `ChargeObjectId`. The charge object identifier shall be unique within the EFC context.

The data element `entryLocationClass` shall contain the location class which is applicable to this particular entry point. The data type shall be `LocationClassId`.

The optional data element `applicableTimeClasses` shall contain a list defining all applicable values for the time classes. The data type of the list elements shall be `TimeClassId`. This data element shall be used to define those time classes which are valid at this specific entry point.

NOTE This option allows the definition of cordon pricing schemes with different time classes at different entry points.

The optional data element **cordonExitLocation** shall be present in case the respective cordon border segment is considered as an exit location of the tolled cordon.

The data element **exitLocationId** shall contain a charge object identifier. The data type shall be **ChargeObjectId**. The charge object identifier shall be unique within the EFC context.

The data element **exitLocationClasses** shall contain a list of location classes. These location classes may also depend on the entry location (shall be contained in an optional list of data elements **entryLocation** of the data type **LocationClassId**). The location classes shall be specified using the data type **LocationClassId**.

The optional data element **applicableTimeClasses** shall contain a list defining all applicable values for the time classes. The data type of the list elements shall be **TimeClassId**. This data element shall be used to define those time classes which are valid at this specific exit point.

8.3.5 Data set “Reporting rules”

The data element in the data set Reporting Rules shall give information regarding

- rules which force the generation of a charge report and
- the content of the charge report.

NOTE The definition of the charge report and its data elements is given in ISO/TS 17575-1 and is therefore outside the scope of this document.

The following EFC attributes belong to the data set Reporting Rules

– **ChargeReportingEvents**

– **ChargeReportConfiguration**

The following clauses give the description and definition of these EFC attributes.

8.3.5.1 EFC attribute ChargeReportingEvents

The attribute **ChargeReportingEvents** shall contain a definition of all events which shall trigger the generation of a charge report in the Front End.

The following event types are supported by this version of the Technical Specification:

– events depending on absolute points in time

– events depending on relative time periods

– events depending on the travelled distance

– events depending on passing specified locations

– events depending on account limits

Structure and data elements of the EFC attribute **ChargeReportingEvents** are given in Table 20 below and defined in Annex A.

Table 20 — EFC attribute `ChargeReportingEvents` (informative)

EFC Attribute	Data element	Data Type	Remark
ChargeReportingEvents	absoluteTimeEvents	SEQUENCE OF AbsoluteTimeEvent	optional
	relativeTimeEvents	SEQUENCE OF RelativeTimeEvent	optional
	travelledDistanceEvents	Distance	optional
	locationEvents	SEQUENCE OF LocationEvent	optional
	feeLimitEvents	FeeLimit	optional
	chargeReportingEventsVersion	VersionAndValidity	
	chargeReportingEventsAuthenticator	MessageAuthenticator	optional

The optional data element **`absoluteTimeEvents`** shall contain a list of the descriptions of events which depend on specified absolute points in time. Each list element shall have the data type **`AbsoluteTimeEvent`**. This data type is described in clause 8.3.5.1.1.

The optional data element **`relativeTimeEvents`** shall contain a list of the descriptions of events which depend on specified lengths of time. Each list element shall have the data type **`RelativeTimeEvent`**. This data type is described in clause 8.3.5.1.2.

The optional data element **`travelledDistanceEvents`** shall contain a distance value. When the vehicle has travelled the specified distance it shall trigger the generation of a charge report. The data type shall be **`Distance`** defined in ISO/TS 17575-1.

The optional data element **`locationEvents`** shall contain a list of the descriptions of events which depend on specified locations. When the vehicle passes the specified location is shall trigger the generation of a charge report. Each element of the list shall have the data type **`LocationEvent`**. This data type is described in clause 8.3.5.1.3.

The optional data element **`feeEvent`** shall contain the definition for trigging a charge report depending on the amount of fee collected since the last charge report has been initiated. The data type shall be **`FeeLimit`** which is of data type **`PaymentFee`** defined in ISO 14906. In case the account balance reaches the value specified in the data element the generation of a charge report shall be triggered.

The data element **`chargeReportEventsVersion`** shall contain version and validity information for the data contained in the EFC attribute **`chargeReportEvents`**. The data type shall be **`VersionAndValidity`**. The data element shall be optional. For details refer to clause 6.2.

The data element **`chargeReportEventsAuthenticator`** shall contain a message authenticator valid for data contained in the EFC attribute **`ChargeReportEvents`**. The data type shall be **`MessageAuthenticator`** defined in ISO/TS 17575-1. The data element **`chargeReportEventsAuthenticator`** shall be optional.

8.3.5.1.1 Data type AbsoluteTimeEvent

The data type **`AbsoluteTimeEvent`** shall define a point in time. This point in time shall be of data type **`Time`**.

Once the point in time is reached the Front End shall trigger the generation of a charge report.

In addition the data type **AbsoluteTimeEvent** shall contain the optional data element **randomDelay**. The value specified in this data element shall delay the generation of the charge report by the given period. The data time shall be **Int4**. The value shall be given in seconds.

8.3.5.1.2 Data type RelativeTimeEvent

The data type **RelativeTimeEvent** shall contain a time interval. The data type shall be **Int3**. The value shall be given in seconds. The start of the interval shall be triggered by the generation of the last Charge Report.

Once the time interval has elapsed the Front End shall trigger the generation of a charge report.

In addition the data type **RelativeTimeEvent** shall contain the optional data element **randomDelay**. The value specified in this data element shall delay the generation of the charge report by the given period. The data time shall be **Int3**. The value shall be given in seconds.

8.3.5.1.3 Data type LocationEvent

The data type **LocationEvent** shall contain a list of locations. Once the vehicle has reached the specified location, the Front End shall trigger the generation of a charge report. The list shall contain the optional location types:

- chargeObject
- line
- area

The optional data element **chargeObject** shall contain the charge object identifier. The data type shall be **ChargeObjectID**.

Once the Front End detects that is has reached the respective charge object it shall trigger the generation of a charge report.

The optional data element **line** shall contain a line description. The data type shall be **Line**. The data elements **crossingDirection1** and **crossingDirection2** define the travel directions which causes the generation of the charge report. Both data elements are of type **BOOLEAN**.

The data element **crossingDirection1** is valid in the following case: The vehicle crosses the line from the left side to the right side. The viewing direction (to define left and right) shall be from the first point of the line definition to the last point of the line definition.

The data element **crossingDirection2** is valid in the following case: The vehicle crosses the line from the right side to the left side. The viewing direction (to define left and right) shall be from the first point of the line definition to the last point of the line definition.

If the value is set to **FALSE** no charge report shall be created. In case the value is set to **TRUE** a charge report shall be generated.

The data element **area** shall contain an area description. The data type of the data element shall be **Polygon**. The definition of the data element **Polygon** and the inner side of the polygon are outlined in clause 8.3.4.2. The data elements **atExit** and **atEntry** (both shall be of data type **BOOLEAN**) shall give information to distinguish between triggering a charge report at exit or entry of the specified area.

8.3.5.2 EFC attribute ChargeReportConfiguration

Charge reports are generated by the Front End. Content, semantic and coding of the charge reports are defined in ISO/TS 17575-1 and are therefore outside the scope of this part of ISO/TS 17575.

As defined in ISO/TS 17575-1 the EFC attribute **ChargeReport** includes optional data elements. It is assumed that the presence of these optional data elements depends on

- type and nature of the toll scheme
- toll charger requirements
- service provider requirements

NOTE More details and considerations regarding the charge report are given in ISO/TS 17575-1.

The attribute **ChargeReportConfiguration** shall be used to actually set the contents of the charge reports which shall be generated by the Front End. ISO/TS 17575-1 defines the data elements of the charge report with respect to their coding and semantics. This part of ISO/TS 17575 defines how the Back End shall request the presence of the optional date elements in the charge report.

To request the presence of an optional data element in the charge report the corresponding data element in the attribute **ChargeReportConfiguration** shall be set to "**TRUE**".

To request the absence of an optional data element in charge report the corresponding data element in the attribute **ChargeReportConfiguration** shall be set to "**FALSE**".

Structure and data elements of the EFC attribute **ChargeReportConfiguration** are given in Table 21 and defined in Annex A.

Table 21 — EFC attribute ChargeReportConfiguration (informative)

EFC Attribute	Data element	Data Type	Remark
ChargeReportConfiguration	chargeReportContent	ChargeReportContent	
	usageStatementContent	UsageStatementContent	
	cccAttributesContent	CCCAttributesContent	optional
	aggregatedSingleTariffClassSession Content	AggregatedSingleTariff ClassSessionContent	optional
	detectedChargeObjectContent	DetectedChargeObject Content	optional
	listOfRawUsagedataContent	ListOfRawUsageData Content	optional
	vehicleDescriptionContent	VehicleDescriptionContent	optional
	chargeReportConfigurationVersion	VersionAndValidity	
	chargeReportConfiguration Authenticator	MessageAuthenticator	optional

The data element **chargeReportConfigurationVersion** shall contain version and validity information for the data contained in EFC attribute **ChargeReportConfiguration**. The data type shall be **VersionAndValidity**. For details refer to clause 6.2.

The data element **chargeReportConfigurationAuthenticator** shall contain a message authenticator valid for data contained in the EFC attribute **chargeReportConfiguration**. The data type shall be **MessageAuthenticator** defined in ISO/TS 17575-1. The data element **chargeReportConfigurationAuthenticator** shall be optional.

8.3.5.2.1 Data type ChargeReportContent

As defined in ISO/TS 17575-1 the data type **chargeReport** includes optional data elements.

The attribute **chargeReportContent** shall contain a list of data elements of the data type BOOLEAN. Each data element in this list is associated to one optional data element in the attribute **chargeReport**.

To request the presence of an optional data element in attribute **chargeReport** the corresponding data element in the attribute **chargeReportContent** shall be set to "**TRUE**".

To request the absence of an optional data element in attribute **chargeReport** the corresponding data element in the attribute **chargeReportContent** shall be set to "**FALSE**".

The following table contains a list of all data elements of the EFC attribute **chargeReportContent**.

Table 22 — Data elements in EFC attribute ChargeReportContent

Data element	Data type (informative)	Description
obeID	BOOLEAN	Requests presence or absence of data element **obeID** in the charge report
paymentMeans	BOOLEAN	Requests the presence or absence of data element **paymentMeans** in the charge report
tollCharger	BOOLEAN	Requests the presence or absence of data element **tollCharger** in the charge report
versionInfo	BOOLEAN	Requests the presence or absence of data element **versionInfo** in the charge report
vatForThisSession	BOOLEAN	Requests the presence or absence of data element **vatForThisSession** in the charge report
accountStatus	BOOLEAN	Requests the presence or absence of data element **accountStatus** in the charge report
transactionCounter	BOOLEAN	Requests the presence or absence of data element **transactionCounter** in the charge report
mileage	BOOLEAN	Requests the presence or absence of data element **mileage** in the charge report
listOfCCCAttributes	BOOLEAN	Requests the presence or absence of data element **listOfCCCAttributes** in the charge report
authenticator	BOOLEAN	Requests the presence of data element **authenticator** in the charge report

8.3.5.2.2 Data type UsageStatementContent

As defined in ISO/TS 17575-1 the data type **`UsageStatement`** includes optional data elements.

The data type **`UsageStatementContent`** shall contain a list of data elements of the data type BOOLEAN. Each data element is this list is associated to one optional data element in the data element **`UsageStatement`** in the attribute **`ChargeReport`**.

To request the presence of an optional data element in data type **`UsageStatement`** the corresponding data element in data type **`UsageStatementContent`** shall be set to "**`TRUE`**".

To request the absence of an optional data element in data type **`UsageStatement`** the corresponding data element in data type **`UsageStatementContent`** shall be set to "**`FALSE`**".

The following table contains a list of all data elements of the data type **`UsageStatementContent`**.

Table 23 — Data elements in data type UsageStatementContent

Data element	Data type (informative)	Description
usageStatementID	BOOLEAN	Requests the presence or absence of data element **`usageStatementID`** in the charge report
regimeID	BOOLEAN	Requests the presence or absence of data element **`regimeID`** in **`usageStatement`** in the charge report
aggregatedFee	BOOLEAN	Requests the presence or absence of data element **`aggregatedFee`** in the charge report
aggregatedSingleClassTariffSession	BOOLEAN	Requests the presence or absence of data element **`aggregatedSingleClassTariffSession`** in the charge report
listOfChargeObjects	BOOLEAN	Requests the presence or absence of data element **`listOfChargeObjects`** in the charge report
listOfDSRCUsageData	BOOLEAN	Requests the presence or absence of data element **`listOfDSRCUsageData`** in the charge report
listOfRawUsageData	BOOLEAN	Requests the presence or absence of data element **`listOfRawUsageData`** in the charge report
noUsage	BOOLEAN	Requests the presence or absence of data element **`noUsage`** in the charge report
usageAuthenticator	BOOLEAN	Requests the presence or absence of data element **`usageAuthenticator`** in the charge report

8.3.5.2.3 Data type CCCAttributesContent

As defined in ISO/TS 17575-1 the data type ccc**`Attributes`** includes optional data elements.

The data type ccc**`AttributesContent`** shall contain a list of data elements of the data type BOOLEAN. Each data element is this list is associated to one optional data element in the data element ccc**`Attributes`** in the attribute **`ChargeReport`**.

To request the presence of an optional data element in data type **cccAttributes** the corresponding data element in data type **cccAttributesContent** shall be set to "**TRUE**".

To request the absence of an optional data element in data type **cccAttributes** the corresponding data element in data type **cccAttributesContent** shall be set to "**FALSE**".

The following table contains a list of all data elements of the data type **cccAttributesContent**.

Table 24 — Data elements in data type CCCAttributesContent

Data element	Data type (informative)	Description
timeOfCCCRecord	BOOLEAN	Requests the presence or absence of data element **timeOfCCCRecord** in data element **CCCAttributes** in the charge report
axlesHistory	BOOLEAN	Requests the presence or absence of data element **axlesHistory** in data element **CCCAttributes** in the charge report
commStatus	BOOLEAN	Requests the presence or absence of data element **commStatus** data element **CCCAttributes** in the charge report
gnssStatus	BOOLEAN	Requests the presence or absence of data element **gnssStatus** in data element **CCCAttributes** in the charge report
distRecStatus	BOOLEAN	Requests the presence or absence of data element **distRecStatus** in data element **CCCAttributes** in the charge report
activeContext	BOOLEAN	Requests the presence or absence of data element **activeContext** in data element **CCCAttributes** in the charge report
obeHistory	BOOLEAN	Requests the presence or absence of data element **obeHistory** in data element **CCCAttributes** in the charge report

8.3.5.2.4 Data type AggregatedSingleTariffClassSessionContent

As defined in ISO/TS 17575-1 the data type **AggregatedSingleTariffClassSession** includes optional data elements.

The data type **AggregatedSingleTariffClassSessionContent** shall contain a list of data elements of the data type BOOLEAN. Each data element is this list is associated to one optional data element in the data element **AggregatedSingleTariffClassSession.**

To request the presence of an optional data element in data type **AggregatedSingleTariffClassSession** the corresponding data element in data type **AggregatedSingleTariffClassSessionContent** shall be set to "**TRUE**".

To request the absence of an optional data element in data type **AggregatedSingleTariffClassSession** the corresponding data element in data type **AggregatedSingleTariffClassSessionContent** shall be set to "**FALSE**".

The following table contains a list of all data elements of the data type **AggregatedSingleTariffClassSessionContent**.

Table 25 — Data elements in data type AggregatedSingleTariffClassSessionContent

Data element	Data type (informative)	Description
timePeriodCovered	BOOLEAN	Requests the presence or absence of data element **timePeriodCovered** in data element **AggregatedSingleTariffClassSession**
vehicleDescription	BOOLEAN	Requests the presence or absence of data element **vehicleDescription** in data element **AggregatedSingleTariffClassSession**
tariffClass	BOOLEAN	Requests the presence or absence of data element **tariffClass** data element **AggregatedSingleTariffClassSession**
totalDistanceCovered	BOOLEAN	Requests the presence or absence of data element **totalDistanceCovered** in data element **AggregatedSingleTariffClassSession**
numberOfDetectedEvents	BOOLEAN	Requests the presence or absence of data element **numberOfDetectedEvents** in data element **AggregatedSingleTariffClassSession**
obeStatus	BOOLEAN	Requests the presence or absence of data element **obeStatus** in data element **AggregatedSingleTariffClassSession**
feeExclVat	BOOLEAN	Requests the presence or absence of data element **feeExclVat** in data element **AggregatedSingleTariffClassSession**
vat	BOOLEAN	Requests the presence or absence of data element **vat** in data element **AggregatedSingleTariffClassSession**

8.3.5.2.5 Data type DetectedChargeObjectContent

As defined in ISO/TS 17575-1 the data type **DetectedChargeObject** includes optional data elements.

The data type **DetectedChargeObjectContent** shall contain a list of data elements of the data type BOOLEAN. Each data element is this list is associated to one optional data element in the data element **DetectedChargeObject.**

To request the presence of an optional data element in data type **DetectedChargeObject** the corresponding data element in data type **DetectedChargeObjectContent** shall be set to "**TRUE**".

To request the absence of an optional data element in data type **DetectedChargeObject** the corresponding data element in data type **DetectedChargeObjectContent** shall be set to "**FALSE**".

The following table contains a list of all data elements of the data type **DetectedChargeObjectContent**.

Table 26 — Data elements in data type DetectedChargeObjectContent

Data element	Data type (informative)	Description
subObjectNumber	BOOLEAN	Requests the presence or absence of data element **subObjectNumber** in data element **DetectedChargeObject**
timeWhenUsed	BOOLEAN	Requests the presence or absence of data element **timeWhenUsed** in data element **DetectedChargeObject**
mileageWhenUsed	BOOLEAN	Requests the presence or absence of data element **mileageWhenUsed** data element **DetectedChargeObject**
vehicleDescription	BOOLEAN	Requests the presence or absence of data element **vehicleDescription** in data element **DetectedChargeObject**
tariffClass	BOOLEAN	Requests the presence or absence of data element **tariffClass** in data element **DetectedChargeObject**
obeStatus	BOOLEAN	Requests the presence or absence of data element **obeStatus** in data element **DetectedChargeObject**
feeExclVat	BOOLEAN	Requests the presence or absence of data element **feeExclVat** in data element **DetectedChargeObject**
vat	BOOLEAN	Requests the presence or absence of data element **vat** in data element **DetectedChargeObject**

8.3.5.2.6 Data type ListOfRawUsageDataContent

As defined in ISO/TS 17575-1 the data type **ListOfRawUsageData** includes optional data elements.

The data type **ListOfRawUsageDataContent** shall contain a list of data elements of the data type BOOLEAN. Each data element is this list is associated to one optional data element in the data element **ListOfRawUsageData.**

To request the presence of an optional data element in data type **ListOfRawUsageData** the corresponding data element in data type **ListOfRawUsageDataContent** shall be set to "**TRUE**".

To request the absence of an optional data element in data type **ListOfRawUsageData** the corresponding data element in data type **ListOfRawUsageDataContent** shall be set to "**FALSE**".

The following table contains a list of all data elements of the data type **ListOfRawUsageDataContent**.

Table 27 — Data elements in data type ListOfRawUsageDataContent

Data element	Data type (informative)	Description
vehicleDescription	BOOLEAN	Requests the presence or absence of data element **vehicleDescription** in data element **ListOfRawUsageData**
tariffClass	BOOLEAN	Requests the presence or absence of data element **tariffClass** in data element **ListOfRawUsageData**

8.3.5.2.7 Data type VehicleDescriptionContent

As defined in ISO/TS 17575-1 the data type **VehicleDescription** includes optional data elements.

The data type **VehicleDescriptionContent** shall contain a list of data elements of the data type BOOLEAN. Each data element is this list is associated to one optional data element in the data element **VehicleDescription.**

To request the presence of an optional data element in data type **VehicleDescription** the corresponding data element in data type **VehicleDescriptionContent** shall be set to "**TRUE**".

To request the absence of an optional data element in data type **VehicleDescription** the corresponding data element in data type **VehicleDescriptionContent** shall be set to "**FALSE**".

The following table contains a list of all data elements of the data type **VehicleDescriptionContent**.

Table 28 — Data elements in data type VehicleDescriptionContent

Data element	Data type (informative)	Description
vehicleLPNr	BOOLEAN	Requests the presence or absence of data element **vehicleLPNr** in data element **VehicleDescription**
axles	BOOLEAN	Requests the presence or absence of data element **axles** in data element **VehicleDescription**
class	BOOLEAN	Requests the presence or absence of data element **class** data element **VehicleDescription**
dimensions	BOOLEAN	Requests the presence or absence of data element **dimensions** in data element **VehicleDescription**
specificCharacters	BOOLEAN	Requests the presence or absence of data element **specificCharacters** in data element **VehicleDescription**
ladenWeight	BOOLEAN	Requests the presence or absence of data element **ladenWeight** in data element **VehicleDescription**
weightLimits	BOOLEAN	Requests the presence or absence of data element **weightLimits** in data element **VehicleDescription**

Annex A
(normative)
EFC data type specifications

A.1 General

The EFC data types and associated coding related to the EFC attributes, data elements and types described in Clauses 6, 7 and 8, are defined using the Abstract Syntax Notation One (ASN.1) technique according to ISO/IEC 8824-1.

The following Figure A.1 and Figure A.2 are included to illustrate the hierarchical structure of the context data elements. These structures are not complete and ignore the lower layers.

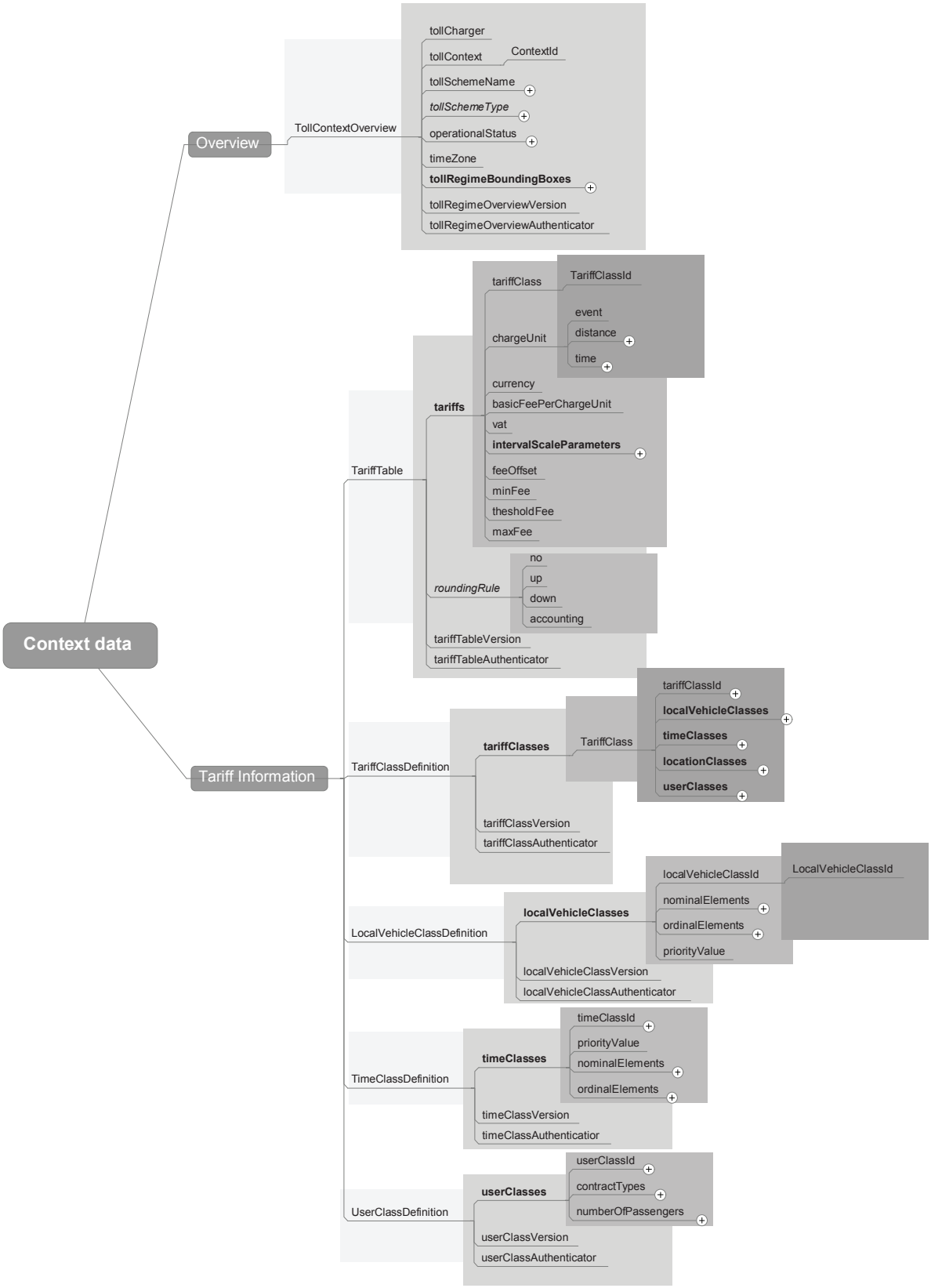


Figure A.1 — Hierarchical structure of context data overview and tariff information (informative)

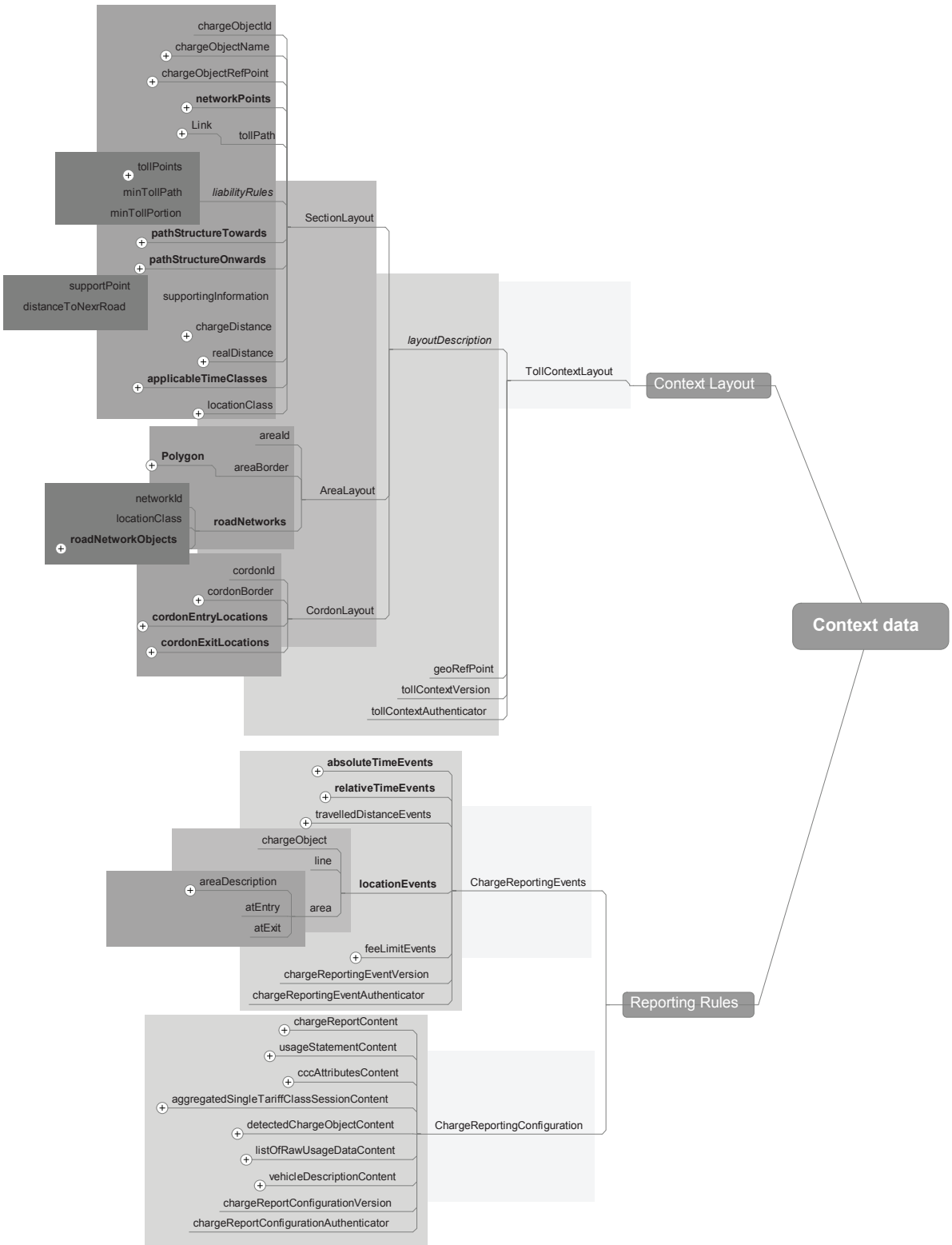


Figure A.2 — Hierarchical structure of context data context layout and reporting rules (informative)

A.2 Data specifications

```
ContextDataModule {iso standard 175753 modules(0) efc(0) version(1)} DEFINITIONS AUTOMATIC
TAGS
::= BEGIN

IMPORTS

TollCharger, Distance, Duration, VersionID
FROM ChargingModule {iso standard 17575 modules(0) efc(0) version(1)}

EquipmentOBUId, PaymentMeans, EFC-ContextMark, Provider, DateAndTime, DateCompact,
PayUnit, VehicleAxles, VehicleClass, VehicleDimensions, VehicleSpecificCharacteristics,
VehicleWeightLaden, VehicleWeightLimits, EnvironmentalCharacteristics, PaymentFee
FROM EfcModule {iso standard 14906 modules(0) efc(0) version(1)}

Longitude, Latitude
FROM CccModule {iso standard 12813 modules(0) efc(0) version(1)};


-- NOTE: The following is the definition of the ISO17575-3 ADU

Iso17575-3Adu ::=     SEQUENCE {
    aduHeader             Iso17575-3AduHeader,
    aduBody               Iso17575-3AduBody
    }

-- NOTE: The following are the definitions of the 17575-3 ADU Header

Iso17575-3AduHeader ::=   SEQUENCE {
    informationSender         Provider,
    informationOriginator     Provider,
    tollCharger               Provider,
    contextId                 ContextId,
    aduSequenceNumber         Int4,
    aduAuthenticator          BIT STRING OPTIONAL
    }

-- NOTE: The following are the definitions of the ISO 17575-3 ADU Body

Iso17575-3AduBody ::=         SEQUENCE {
    tollContextOverview           TollContextOverview OPTIONAL,
    tariffTable                   TariffTable OPTIONAL,
    tariffClassDefinition         TariffClassDefinition OPTIONAL,
    localVehicleClassDefinition   LocalVehicleClassDefinition OPTIONAL,
    timeClassDefinition           TimeClassDefinition OPTIONAL,
    userClassDefinition           UserClassDefinition OPTIONAL,
    tollContextLayout             TollContextLayout OPTIONAL,
    chargeReportingEvents         ChargeReportingEvents OPTIONAL,
    chargeReportConfiguration     ChargeReportConfiguration OPTIONAL
    }
```

```
-------------------------------------------------------------------------------------
-- NOTE: The following are the definitions of the EFC Attributes
-------------------------------------------------------------------------------------

TollContextOverview ::=          SEQUENCE {
    tollCharger                      TollCharger,
    tollContext                      ContextId,
    tollSchemeName                   UTF8String OPTIONAL,
    tollSchemeType                   TollSchemeType,
    operationalStatus                OperationalStatus,
    timeZone                         INTEGER (-720..720),
    tollContextBoundingBoxes         SEQUENCE OF SphericalBox OPTIONAL,
    tollContextOverviewVersion       VersionAndValidity,
    tollContextOverviewAuthenticator MessageAuthenticator OPTIONAL
    }

TariffTable ::=                  SEQUENCE {
    tariffs                          SEQUENCE OF Tariff,
    roundingRule                     RoundingRule OPTIONAL,
    tariffTableVersion               VersionAndValidity,
    tariffTableAuthenticator         MessageAuthenticator OPTIONAL
    }

TariffClassDefinition ::=            SEQUENCE {
    tariffClasses                        SEQUENCE OF TariffClass,
    tariffClassDefinitionVersion         VersionAndValidity,
    tariffClassDefinitionAuthenticator   MessageAuthenticator OPTIONAL
    }

LocalVehicleClassDefinition ::=  SEQUENCE {
    localVehicleClasses              SEQUENCE OF LocalVehicleClass,
    localVehicleClassVersion         VersionAndValidity,
    localVehicleClassAuthenticator   MessageAuthenticator OPTIONAL
    }

TimeClassDefinition ::=          SEQUENCE {
    timeClasses                      SEQUENCE OF TimeClass,
    timeClassDefinitionVersion       VersionAndValidity,
    timeClassDefinitionAuthenticator MessageAuthenticator OPTIONAL
    }

UserClassDefinition ::=          SEQUENCE {
    userClasses                      SEQUENCE OF UserClass,
    userClassDefinitionVersion       VersionAndValidity,
    userClassDefinitionAuthenticator MessageAuthenticator OPTIONAL
    }

TollContextLayout ::=     SEQUENCE {
    layoutDescription         Layout,
    geoRefPoint               Point OPTIONAL,
    tollContextVersion        VersionAndValidity,
    tollContextAuthenticator  MessageAuthenticator OPTIONAL
    }

ChargeReportingEvents ::=            SEQUENCE {
    absoluteTimeEvents                   SEQUENCE OF AbsoluteTimeEvent OPTIONAL,
    relativeTimeEvents                   SEQUENCE OF RelativeTimeEvent OPTIONAL,
    travelledDistanceEvents              Distance OPTIONAL,
    locationEvents                       SEQUENCE OF LocationEvent OPTIONAL,
    feeLimitEvents                       FeeLimit OPTIONAL,
    chargeReportingEventsVersion         VersionAndValidity,
    chargeReportingEventsAuthenticator   MessageAuthenticator OPTIONAL
    }
```

```
ChargeReportConfiguration ::=            SEQUENCE {
    chargeReportContent                      ChargeReportContent,
    usageStatementContent                    UsageStatementContent,
    cccAttributesContent                     CccAttributesContent OPTIONAL,
    aggregatedSingleTariffClassSessionContent    AggregatedSingleTariffClassSessionContent
                                                 OPTIONAL,
    detectedChargeObjectContent              DetectedChargeObjectContent OPTIONAL,
    listOfRawUsageDataContent                ListOfRawUsageDataContent OPTIONAL,
    vehicleDescriptionContent                VehicleDescriptionContent OPTIONAL,
    chargeReportConfigurationVersion         VersionAndValidity,
    chargeReportConfigurationAuthenticator   MessageAuthenticator OPTIONAL
    }


-------------------------------------------------------------------------------
-- NOTE: The following are the definitions of the data elements in alphabetic order
-------------------------------------------------------------------------------


AbsolutePointCoordinates ::=         SEQUENCE {
    longitude                            Longitude,
    latitude                             Latitude
    }

AbsoluteTimeEvent ::=         SEQUENCE {
    timesDuringDay                Time,
    randomDelay                   Int3 OPTIONAL
    }

ActualNumberOfPassengers ::=  Int1
    -- according to ISO 14906
    -- actual number of passengers (i.e. human beings) present in the vehicle,
    -- incl. the driver

AggregatedSingleTariffClassSessionContent ::=   SEQUENCE {
    timePeriodCovered         BOOLEAN,
    vehicleDescription        BOOLEAN,
    tariffClass               BOOLEAN,
    totalDistanceCovered      BOOLEAN,
    numberOfDetectedEvents    BOOLEAN,
    obeStatus                 BOOLEAN,
    feeExclVat                BOOLEAN,
    vat                       BOOLEAN
    }

AreaId ::= Int1

AreaLayout ::=                SEQUENCE {
    areaId                    AreaId,
    aeaBorder                 Polygon,
    locationClass             LocationClassId,
    roadNetworks              SEQUENCE OF RoadNetwork OPTIONAL
    }

CccAttributesContent ::=  SEQUENCE {
    timeOfCCCRecord           BOOLEAN,
    axlesHistory              BOOLEAN,
    commStatus                BOOLEAN,
    gnssStatus                BOOLEAN,
    distRecStatus             BOOLEAN,
    activeContext             BOOLEAN,
    obeHistory                BOOLEAN
    }
```

```
ChargeDistance ::=            CHOICE {
    predefinedDistance            Distance,
    entryDependingDistance        SEQUENCE {
        entryDependingDistance        SEQUENCE OF EntryDependingDistanceData,
        missingEntryInformation       Distance
        }
    }

CO2EmissionValue ::= Int2 --according to vehicle registration documents in g/km

CO2EmissionValueRange ::= SEQUENCE {
    lowerLimit                CO2EmissionValue,
    upperLimit                CO2EmissionValue
    }

ChargeObjectId ::= Int4

ChargeReportContent ::=   SEQUENCE {
    obeId                     BOOLEAN,
    paymentMeans              BOOLEAN,
    tollCharger               BOOLEAN,
    versionInfo               BOOLEAN,
    vatForThisSession         BOOLEAN,
    accountStatus             BOOLEAN,
    transactionCounter        BOOLEAN,
    mileage                   BOOLEAN,
    listOfCCCAttributes       BOOLEAN,
    authenticator             BOOLEAN
    }

ChargeUnit ::=        CHOICE {
    distance              Distance,
    time                  Duration,
    event                 Int1
    }

ContextId ::= EntityId

ContractTypes ::=     SEQUENCE {
    contractProvider      Provider,
    typeOfContract        OCTET STRING(SIZE(2))
    }

CopValue ::=       ENUMERATED { -- same definition as in ISO 14906
    noEntry           (0),
    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
    }

CopValueRange ::=     SEQUENCE {
    lowerLimit            CopValue,
    upperLimit            CopValue
    }

CordonBorderSegment ::=   SEQUENCE {
    cordonSegmentId           CordonSegmentId,
    startPoint                Point,
    cordonEntryLocation       CordonEntryLocation OPTIONAL,
    cordonExitLocation        CordonExitLocation OPTIONAL
    }
```

```
CordonEntryLocation ::=       SEQUENCE {
    entryLocationId               ChargeObjectId,
    entryLocationClass            LocationClassId,
    applicableTimeClasses         SEQUENCE OF TimeClassId OPTIONAL
    }

CordonExitLocation ::=        SEQUENCE {
    exitLocationId                ChargeObjectId,
    exitLocationClass             SEQUENCE OF SEQUENCE {
        locationClass                 LocationClassId,
        entryLocation                 SEQUENCE OF EntryLocationId OPTIONAL
        },
    applicableTimeClasses         SEQUENCE OF TimeClassId OPTIONAL
    }

CordonId ::= Int1

CordonLayout ::=              SEQUENCE {
    cordonId                      CordonId,
    cordonBorderPolygon           SEQUENCE OF CordonBorderSegment
    }

CordonSegmentId ::=   Int4

Currency ::=   INTEGER(1..999)    -- Currency code according to ISO 4217

DetectedChargeObjectContent ::=  SEQUENCE {
    subObjectNumber                   BOOLEAN,
    timeWhenUsed                      BOOLEAN,
    mileageWhenUsed                   BOOLEAN,
    vehicleDescription                BOOLEAN,
    tariffClass                       BOOLEAN,
    obeStatus                         BOOLEAN,
    feeExclVat                        BOOLEAN,
    vat                               BOOLEAN
    }

EntityId ::=   Provider

EntryDependingDistanceData ::=    SEQUENCE {
    chargeObjectAtEntry               ChargeObjectId,
    definedDistance                   Distance
    }

EntryLocationId ::=   Int4

EuroValue ::=     ENUMERATED { --same definition as in ISO 14906
    noEntry           (0),
    euro-1            (1),
    euro-2            (2),
    euro-3            (3),
    euro-4            (4),
    euro-5            (5),
    euro-6            (6),
    reservedForUse1   (7)
    }

EuroValueRange ::=    SEQUENCE {
    lowerLimit                EuroValue,
    upperLimit                EuroValue
    }

FeeLimit ::= PaymentFee

Int1 ::= INTEGER(0..255)

Int2 ::= INTEGER(0..65535)
```

```
Int2Signed ::= INTEGER(-32768..32767)

Int3 ::= INTEGER(0..16777215)

Int4 ::= INTEGER(0..4294967295)

IntervalScaleParameter ::=          SEQUENCE {
    vehicleMaxLadenWeightIntervals      SEQUENCE {
        zeroOffset                          Int2Signed,
            -- same definition as in ISO 14906 in attribute VehicleMaxLadenWeight
        resolution                          VehicleMaxLadenWeight,
        max                                 VehicleMaxLadenWeight
        } OPTIONAL,
    vehicleLengthOverallIntervals       SEQUENCE {
        zeroOffset                          Int2Signed,
            -- same definition as in ISO 14906 in attribute VehicleLengthOverall
        resolution                          VehicleLengthOverall,
        max                                 VehicleLengthOverall
        } OPTIONAL
    }

Layout ::=                CHOICE {
    sectionPricingLayout      SEQUENCE OF SectionLayout,
    areaPricingLayout         SEQUENCE OF AreaLayout,
    cordonPricingLayout       SEQUENCE OF CordonLayout
    }

LiabilityRules ::=        CHOICE {
    tollPoints                SEQUENCE(SIZE(1..2)) OF Point,
    minTollPath               Link,
    minTollPortion            INTEGER(0..1000)   -- value in 1‰
    }

Line ::=       SEQUENCE {
    point1         Point,
    point2         Point
    }

Link ::=                      SEQUENCE {
    linkId                        LinkId,
    startPoint                    Point,
    endPoint                      Point,
    intermediatePoints            SEQUENCE OF Point OPTIONAL
    }

LinkId ::= Int4

ListOfRawUsageDataContent ::= SEQUENCE {
    vehicleDescription            BOOLEAN,
    tariffClass                   BOOLEAN
    }

LocalVehicleClass ::=         SEQUENCE {
    localVehicleClassId               LocalVehicleClassId,
    nominalElements                   NominalVehicleParameters,
    ordinalElements                   OrdinalVehicleParameters OPTIONAL,
    priorityValue                     Int1 OPTIONAL -- 0: lowest priority, 255: highest
    priority
    }

LocalVehicleClassId ::= Int1

LocationClassId ::= Int4
```

```
LocationEvent ::=          SEQUENCE {
    chargeObject               ChargeObjectId OPTIONAL,
    line                       SEQUENCE {
        lineDescription            Line,
        crossingDirection1         BOOLEAN,
        crossingDirection2         BOOLEAN
        } OPTIONAL,
    area                       SEQUENCE {
        areaDescription            Polygon,
        atEntry                    BOOLEAN,
        atExit                     BOOLEAN
        } OPTIONAL
    }

MessageAuthenticator ::= BIT STRING

NominalTimeParameters ::=      SEQUENCE {
    weekdays                       SEQUENCE OF Weekday OPTIONAL,
    dates                          SEQUENCE OF DateCompact OPTIONAL,
    classesSetExternally           SEQUENCE OF ENUMERATED {
        congestionChargeLevel1         (1),
        congestionChargeLevel2         (2),
        futureUse1                     (3),
        futureUse2                     (4),
        futureUse3                     (5),
        futureUse4                     (6),
        futureUse5                     (7),
        futureUse6                     (8)
        } OPTIONAL
    }

NominalVehicleParameters ::=  SEQUENCE {
    vehicleClasses                 SEQUENCE OF VehicleClass, -- use according to EN 15509
    vehicleAxlesNumbers            SEQUENCE OF VehicleAxles OPTIONAL,
        -- use according to ISO 14906
    euroValues                     SEQUENCE OF EuroValue OPTIONAL,
        -- use according to ISO 14906
    ...
    }

MaxFee ::=      SEQUENCE {
    perDay          Int3 OPTIONAL,     -- in currencies minor units
    perWeek         Int3 OPTIONAL,     -- in currencies minor units
    perMonth        Int3 OPTIONAL,     -- in currencies minor units
    perYear         Int3 OPTIONAL      -- in currencies minor units
    }

OperationalStatus ::=      SEQUENCE {
    startsOperationAt          DateAndTime,
    stopsOperationAt           DateAndTime OPTIONAL
    }
```

```
OrdinalTimeParameters ::=     SEQUENCE {
    weekdays                      SEQUENCE OF SEQUENCE {
        startDay                      Weekday,
        endDay                        Weekday
        } OPTIONAL,
    absoluteTimeOfDay             SEQUENCE OF SEQUENCE {
        startTime                     Time,
        endTime                       Time
        } OPTIONAL,
    relativeTimePeriods           SEQUENCE OF SEQUENCE {
        minPeriod                     Int2 OPTIONAL, -- in minutes
        maxPeriod                     Int2 OPTIONAL -- in minutes
        } OPTIONAL,
    periodsInYear                 SEQUENCE OF SEQUENCE {
        startDay                      DateCompact,
        endDay                        DateCompact
        } OPTIONAL
    }

OrdinalVehicleParameters ::=  SEQUENCE {
    vehicleLengthOverall          SEQUENCE OF VehicleLengthOverallRange OPTIONAL,
    vehicleHeightOverall          SEQUENCE OF VehicleHeightOverallRange OPTIONAL,
    vehicleWidthOverall           SEQUENCE OF VehicleWidthOverallRange OPTIONAL,
    vehicleFirstAxleHeight        SEQUENCE OF VehicleFirstAxleHeightRange OPTIONAL,
    vehicleAxlesNumber            SEQUENCE OF VehicleAxlesNumberRange OPTIONAL,
    vehicleMaxLadenWeight         SEQUENCE OF VehicleMaxLadenWeightRange OPTIONAL,
    vehicleTrainMaximumWeight     SEQUENCE OF VehicleTrainMaximumWeightRange OPTIONAL,
    vehicleWeightUnladen          SEQUENCE OF VehicleWeightUnladenRange OPTIONAL,
    vehicleWeightLaden            SEQUENCE OF VehicleWeightLadenRange OPTIONAL,
    euroValue                     SEQUENCE OF EuroValueRange OPTIONAL,
    copValue                      SEQUENCE OF CopValueRange OPTIONAL,
    vehicleClass                  SEQUENCE OF VehicleClassRange OPTIONAL,
    co2EmissionValue              SEQUENCE OF CO2EmissionValue OPTIONAL,
    ...
    }

Point ::=      CHOICE {
    pointIdentifier               PointId,
    absolutePointCoordinates      AbsolutePointCoordinates,
    relativePointCoordinates      RelativePointCoordinates
    }

PointId ::=    Int4

PointIdDefinition ::= SEQUENCE {
    pointId               PointId,
    pointCoordinates      AbsolutePointCoordinates
    }

Polygon ::= SEQUENCE OF Point

RelativePointCoordinates ::=         SEQUENCE {
    longitude                            Int2Signed,-- in microdegrees
    latitude                             Int2Signed -- in microdegrees
    }

RelativeTimeEvent ::=         SEQUENCE {
    timeDuration                  Int3,
    randomDelay                   Int3 OPTIONAL
    }

RoadNetwork ::=               SEQUENCE {
    networkId                     Int1,
    locationClass                 LocationClassId,
    roadNetworkObjects            SEQUENCE OF RoadNetworkObject
    }
```

```
RoadNetworkObject ::=         SEQUENCE {
    chargeObjectId                ChargeObjectId,
    chargeObjectName              CHARACTER STRING OPTIONAL,
    chargeObjectRefPoint          Point OPTIONAL,
    networkPoints                 SEQUENCE OF Point OPTIONAL,
    tollPath                      Link,
    supportingInformation         SEQUENCE OF SupportingPoint OPTIONAL,
    applicableTimeClasses         SEQUENCE OF TimeClassId OPTIONAL
    }

RoundingRule ::=      ENUMERATED {
    no                      (0), -- no rounding
    up                      (1),
        -- always rounding up to next larger value of the minor curency unit
    down                    (2),
        -- always rounding down to next lower value of the minor currency unit
    accounting              (3),
        -- rounding according to accounting rules as defined in DIN 1333
    futureUse1              (4),
    futureUse2              (5),
    futureUse3              (6),
    futureUse4              (7)
    }

SectionLayout ::=             SEQUENCE {
    chargeObjectId                ChargeObjectId,
    chargeObjectName              UTF8String OPTIONAL,
    chargeObjectRefPoint          Point OPTIONAL,
    networkPoints                 SEQUENCE OF PointIdDefinition OPTIONAL,
    tollPath                      Link,
    liabilityRules                LiabilityRules,
    pathStructureTowards          SEQUENCE OF Link OPTIONAL,
    pathStructureOnwards          SEQUENCE OF Link OPTIONAL,
    supportingInformation         SEQUENCE OF SupportingPoint OPTIONAL,
    chargeDistance                ChargeDistance,
    realDistance                  Distance OPTIONAL,
    applicableTimeClasses         SEQUENCE OF TimeClassId OPTIONAL,
    locationClass                 LocationClassId,
    storageRequired               BOOLEAN OPTIONAL
    }

SphericalBox ::=          SEQUENCE {
    southernLongitude         Longitude,
    northernLongitude         Longitude,
    westernLatitude           Latitude,
    easternLatitude           Latitude
    }

SupportingPoint ::=       SEQUENCE {
    supportPoint              Point,
    distanceToNextRoad        Distance OPTIONAL
    }
```

```
Tariff ::=                    SEQUENCE {
    tariffClass                   TariffClassId,
    chargeUnit                    ChargeUnit,
    currency                      Currency,
    basicFeePerChargeUnit         REAL,
        -- in major units of the defined currency
    vat                           Int3 OPTIONAL,
        -- in 0,01%
    intervalScaleParameters       SEQUENCE OF IntervalScaleParameter OPTIONAL,
    offsetFee                     Int3 OPTIONAL,
        -- in 1/100 of the minor unit of the defined currency
    minFee                        Int3 OPTIONAL,
        -- in 1/100 of the minor unit of the defined currency
    thresholdFee                  Int3 OPTIONAL,
        -- in 1/100 of the minor unit of the defined currency
    maxFee                        MaxFee OPTIONAL
    }

TariffClass ::=           SEQUENCE {
    tariffClassId             TariffClassId,
    localVehicleClasses       SEQUENCE OF LocalVehicleClassId,
    timeClasses               SEQUENCE OF TimeClassId OPTIONAL,
    locationClasses           SEQUENCE OF LocationClassId OPTIONAL,
    userClasses               SEQUENCE OF UserClassId OPTIONAL
    }

TariffClassId ::= Int2

Time ::=       SEQUENCE {
    hours          INTEGER (0..23),
    mins           INTEGER (0..59),
    secs           INTEGER (0..59)
    }

TimeClass ::=          SEQUENCE {
    timeClassId               TimeClassId,
    nominalElements           NominalTimeParameters OPTIONAL,
    ordinalElements           OrdinalTimeParameters OPTIONAL,
    priorityValue             Int1 OPTIONAL -- 0: lowest priority, 255: highest priority
    }

TimeClassId ::= Int1

TollSchemeType ::=        ENUMERATED {
    roadSectionPricing        (0),
    areaPricingDistance       (1),
    areaPricingTime           (2),
    cordonPricing             (3)
    }

UsageStatementContent ::=                SEQUENCE {
    usageStatementId                         BOOLEAN,
    regimeId                                 BOOLEAN,
    aggregatedFee                            BOOLEAN,
    aggregatedSingleTariffClassSession       BOOLEAN,
    listOfChargeObjects                      BOOLEAN,
    listOfDSRCUsageData                      BOOLEAN,
    listOfRawUsageData                       BOOLEAN,
    noUsage                                  BOOLEAN,
    usageAuthenticator                       BOOLEAN
    }

UserClass ::=                 SEQUENCE {
    userClassId                   UserClassId,
    contractTypes                 ContractTypes OPTIONAL,
    actualNumberOfPassengers      ActualNumberOfPassengers OPTIONAL
    }
```

```
UserClassId::= Int1

VehicleAxlesNumberRange ::=       SEQUENCE {
    lowerLimit                        VehicleAxlesNumber, -- according to ISO 14906
    upperLimit                        VehicleAxlesNumber -- according to ISO 14906
    }

VehicleClassRange ::=             SEQUENCE {
    lowerLimit                        VehicleClass, -- according to EN 15509
    upperLimit                        VehicleClass   -- according to EN 15509
    }

VehicleDescriptionContent ::=     SEQUENCE {
    vehicleLPNr                       BOOLEAN,
    axles                             BOOLEAN,
    class                             BOOLEAN,
    dimensions                        BOOLEAN,
    specificCharacters                BOOLEAN,
    ladenWeight                       BOOLEAN,
    weightLimits                      BOOLEAN
    }

VehicleFirstAxleHeightRange ::=   SEQUENCE {
    lowerLimit                        VehicleFirstAxleHeight,
    upperLimit                        VehicleFirstAxleHeight
    }

VehicleHeightOverallRange ::= SEQUENCE {
    lowerLimit                        VehicleHeightOverall,
    upperLimit                        VehicleHeightOverall
    }

VehicleLengthOverallRange ::= SEQUENCE {
    lowerLimit                        VehicleLengthOverall,
    upperLimit                        VehicleLengthOverall
    }

VehicleMaxLadenWeightRange ::=    SEQUENCE {
    lowerLimit                        VehicleMaxLadenWeight,
    upperLImit                        VehicleMaxLadenWeight
    }

VehicleTrainMaximumWeightRange ::=    SEQUENCE {
    lowerLimit                           VehicleTrainMaximumWeight,
    upperLimit                           VehicleTrainMaximumWeight
    }

VehicleWeightUnladenRange ::= SEQUENCE {
    lowerLimit                        VehicleWeightUnladen,
    upperLimit                        VehicleWeightUnladen
    }

VehicleWeightLadenRange ::=       SEQUENCE {
    lowerLimit                        VehicleWeightLaden,
                                                -- accoring to ISO 14906
    upperLimit                        VehicleWeightLaden
                                                -- according to ISO 14906
    }

VehicleWidthOverallRange ::=      SEQUENCE {
    lowerLimit                        VehicleWidthOverall,
    upperLimit                        VehicleWidthOverall
    }
```

```
VehicleLengthOverall ::= Int1
    -- same definition as in ISO 14906 in VehicleDimensions:
    -- Nominal maximum overall length of the vehicle according to ISO 612,
    -- in dm, rounded to the next dm

VehicleHeightOverall ::= Int1
    -- same definition as in ISO 14906 in VehicleDimensions:
    -- Nominal overall unladen height, according to ISO 612,
    -- in dm, rounded to the next dm.

VehicleWidthOverall ::= Int1
    -- same definition as in ISO 14906 in VehicleDimensions:
    -- Nominal overall width, according to ISO 612, in dm, rounded to the next dm

VehicleMaxLadenWeight ::= Int2
    -- same definition as in ISO 14906 in VehicleWeightLimits:
    -- Maximum permissible total weight including payload, according to ISO 1176,
    -- 10 kg units, rounded down to the next 10 kg step.

VehicleTrainMaximumWeight ::= Int2
    -- same definition as in ISO 14906 in VehicleWeightLimits:
    -- Maximum permissible weight of the complete vehicle train, as defined in ISO 1176,
    -- 10 kg units, rounded down to the next 10 kg step

VehicleWeightUnladen ::= Int2
    -- same definition as in ISO 14906 in VehicleWeightLimits:
    -- Nominal unladen weight, according to ISO 1176,
    -- 10 kg units, rounded down to the next 10 kg step

VehicleFirstAxleHeight ::= Int1
    -- same definition as in ISO 14906 in VehicleAxles:
    -- bonnet height, measured over the front axle, in dm, rounded to the next dm

VehicleAxlesNumber ::= Int1
    -- same definition as in ISO 14906 in VehicleAxles:
    -- tyre type and number of axles, including drop axles
    -- for details refer to ISO 14906

VersionAndValidity ::= SEQUENCE {
    version                VersionID,
    validFrom              DateAndTime
    }

Weekday ::=         ENUMERATED {
    monday              (1),
    tuesday             (2),
    wednesday           (3),
    thursday            (4),
    friday              (5),
    saturday            (6),
    sunday              (7),
    notUsed             (8)
    }

END
```

Annex B
(normative)
PICS proforma for the attributes

B.1 Introduction

This clause contains the Protocol Implementation Conformance Statements (PICS) proforma to be used for Front End implementation of the context description defined in clauses 6, 7 and 8 and in Annex A.

B.2 General

To evaluate conformance of a particular implementation, it is necessary to have a statement of which capabilities and options have been implemented. Such a statement is called an Implementation Conformance Statement (ICS) or more specific in case it covers transactions a Protocol Implementation Conformance Statements (PICS). This annex provides PICS templates, to be filled in by equipment suppliers.

B.3 Guidance and structure

The purpose of this PICS proforma is to provide a mechanism whereby a supplier of an implementation of the requirements defined in this part of ISO/TS 17575 may provide information about the implementation in a standardized manner.

The PICS proforma is subdivided into clauses for the following categories of information:

- identification of the implementation;
- identification of the protocol;
- global statement of conformance;
- PICS proforma tables.

B.4 Instruction for completing the PICS proforma

B.4.1 Definition of support

A capability is said to be supported if the Implementation Under Test (IUT) is able:

- to generate the corresponding operation parameters (either automatically or because the end user requires that capability explicitly);
- to interpret, handle and when required make available to the end user the corresponding error or result.

A protocol element is said to be supported for a sending implementation if it is able to generate it under some circumstances (either automatically or because the end user requires relevant services explicitly).

A protocol element is said to be supported for a receiving implementation if it is correctly interpreted and handled and also, when appropriate, made available to the end user.

B.4.2 Status column

This column indicates the level of support required for conformance to this part of ISO/TS 17575. The values are as follows:

m mandatory support is required;

o optional support is permitted for conformance to the standard. If implemented it must conform to the specifications and restrictions contained in the standard. These restrictions may affect the optionality of other items;

c the item is conditional (support of the capability is subject to a predicate);

c: m the item is mandatory if the predicate is true, optional otherwise;

- the item is not applicable;

I the item is outside the scope of this PICS.

In the PICS proforma tables, every leading item marked “m” shall be supported by the IUT. Sub-items marked “m” shall be supported if the corresponding leading item is supported by the IUT.

B.4.3 Support column

This column shall be completed by the supplier or implementer to indicate the level of implementation of each item. The proforma has designed such that values required are:

Y yes, the item has been implemented;

N no, the item has not been implemented;

- the item is not applicable;

All entries within the PICS proforma shall be made in ink. Alterations to such entries shall be made by crossing out, not erasing nor making the original entry illegible, and writing the new entry alongside. All such alterations to records shall be initialized by the staff making them.

B.4.4 Item reference numbers

Each line within the PICS proforma which requires implementation details to be entered is numbered at the left hand edge of the line. This numbering is included as a means of uniquely identifying all possible implementation details within the PICS proforma. This referencing is used both inside the PICS proforma, and for references from other test specification documents.

The means of referencing individual responses is done by the following sequence:

– a reference to the smallest enclosing the relevant item;

– a solidus character, '/';

– the reference number of the row in which the response appears;

– if, and only if, more than one response occurs in the row identified by the reference number, then each possible entry is implicitly labeled a, b, c, etc. from left to right, and this letter is appended to the sequence

B.5 PICS proforma for the Front End

B.5.1 Identification of the implementation

B.5.1.1 Identification of PICS

Item No	Question	Response
1	Date of statement (DD/MM/YY)	
2	PICS Serial Number	
3	System Conformance Statement Cross Reference	

B.5.1.2 Identification of the implementation and/or system

Item No	Question	Response
1	Service provider or EFC context name	
2	Service provider and EFC context identifier	
3	Version number	
4	Other information	

B.5.1.3 Identification of the Front End supplier

Item No	Question	Response
1	Organization name	
2	Contact name(s)	
3	Address	
4	Phone number(s)	
5	e-mail address	
6	Other information	

B.5.1.4 Identification of the Front End

Item No	Question	Response
1	Brand name	
2	Type, version	
3	Manufacturer ID	
4	Equipment class	
5	Serial number(s)	
6	Other information	

B.5.2 Identification of the standard

Item No	Question	Response
1	Title, Reference No, publication date of the Technical Specification	
2	Technical Specification version	
3	Implemented addenda	
4	Implementer's guide version number	
5	Implementation defect reports	
6	Other information	

B.5.3 Global statement of conformance

Are all mandatory capabilities implemented? (Yes/No) ……

NOTE 1 Answering “No” to this question indicates non-conformance to the specification. Non-supported mandatory capabilities are to be identified in the PICS, with an explanation of why the implementation is non-conforming, on pages attached to the PICS proforma.

B.5.4 PICS proforma tables

This part of the PICS proforma identifies the supported application context, the communication services and attributes (ADUs).

B.5.4.1 ADU and ADU Header

Item No	Element	Status	Support
1	Iso17575-3Adu	m	
2	AduAuthenticator	o	

B.5.4.2 Communication services support

Item No	Element	Status	Support
1	Iso17575-2 Communication services supported	m	

B.5.4.3 EFC Attributes

Item No	Element	Status	Support
1	TollContextOverview	m	
2	TariffTable	m	
3	TariffClassDefinition	m	
4	LocalVehicleClassDefinition	m	
5	TimeClassDefinition	o	
6	UserClassDefinition	o	
7	TollContextLayout	m	
8	ChargeReportingEvents	m	
9	ChargeReportConfiguration	m	

B.5.4.4 Toll Context overview

Item No	Element	Status	Support
1	tollSchemeName	o	
2	tollContextBoundingBoxes	o	
3	TollContextOverviewAuthenticator	o	

B.5.4.5 Toll scheme types

Item No	TollSchemeType	Status	Support
1	roadSectionPricing	o	
2	areaPricingDistance	o	
3	areaPricingTime	o	
4	cordonPricing	o	

B.5.4.6 Operational status

Item No	TollRegimeType	Status	Support
1	stopsOperationAt	o	

B.5.4.7 Tariff table and tariffs

Item No	Element	Status	Support
1	vat	o	
2	intervalScaleParameters	o	
3	offsetFee	o	
4	minFee	o	
5	thresholdFee	o	
6	maxFee	o	
7	roundingRule	o	
8	tariffTableAuthenticator	o	

B.5.4.8 Tariff class definitions and tariff classes

Item No	Element	Status	Support
1	localVehicleClasses	m	
2	timeClasses	o	
3	locationClasses	o	
4	userClasses	o	
5	tariffClassDefinitionAuthenticator	o	

B.5.4.9 Local vehicle class definitions and local vehicle classes

Item No	Element	Status	Support
1	ordinalElements	o	
2	priorityValue	o	
3	localVehicleClassDefinitionAuthenticator	o	

B.5.4.10 Nominal vehicle parameters

Item No	Element	Status	Support
1	vehicleClasses	m	
2	vehicleAxlesNumbers	o	
3	euroValues	o	

B.5.4.11 Ordinal vehicle parameters

Item No	Element	Status	Support
1	vehicleLengthOverall	o	
2	vehicleHeightOverall	o	
3	vehicleWidthOverall	o	
4	vehicleFirstAxlesHeight	o	
5	vehicleAxlesNumber	o	
6	vehicleMaxLadenWeight	o	
7	vehicleTrainMaximumWeight	o	
8	vehicleWeightUnladen	o	
9	vehicleWeightLaden	o	
10	euroValue	o	
11	copValue	o	
12	vehicleClass	o	
13	co2EmissionValue	o	

B.5.4.12 Time class definitions and time classes

Item No	Element	Status	Support
1	nominalElements	o	
2	ordinalElements	o	
3	priorityValue	o	
4	timeClassDefinitionAuthenticator	o	

B.5.4.13 Nominal time class parameters

Item No	Element	Status	Support
1	weekdays	o	
2	dates	o	
3	classesSetExternally	o	

B.5.4.14 Ordinal time class parameters

Item No	Element	Status	Support
1	weekdays	o	
2	absoluteTimeOfDay	o	
3	relativeTimePeriods	o	
4	periodsInYear	o	

B.5.4.15 User class definitions and user classes

Item No	Element	Status	Support
1	contractTypes	o	
2	actualNumberOfPassengers	o	
3	userClassDefinitionAuthenticator	o	

B.5.4.16 Toll context layout

Item No	Element	Status	Support
1	geoRefPoint	o	
2	tollContextAuthenticator	o	

B.5.4.17 Supported context layout types

Item No	Element	Status	Support
1	sectionPricingLayout	o	
2	areaPricingLayout	o	
3	cordonPricingLayout	o	

B.5.4.18 Section pricing layout description

Item No	Element	Status	Support
1	chargeObjectName	o	
2	chargeObjectRefPoint	o	
3	networkPoints	o	
4	pathStructureTowards	o	
5	pathStructureOnwards	o	
6	supportingInformation	o	
7	realDistance	o	
8	applicableTimeClasses	o	
9	storageRequired	o	

B.5.4.19 Point

Item No	Element	Status	Support
1	pointIdentifier	o	
2	absolutePointCoordinates	o	
3	relativePointCoordinates	o	

B.5.4.20 Link

Item No	Element	Status	Support
1	intermediatePoints	o	

B.5.4.21 Supporting Point

Item No	Element	Status	Support
1	distanceToNextRoad	o	

B.5.4.22 Area pricing layout description

Item No	Element	Status	Support
1	roadNetwork	o	

B.5.4.23 Road network objects in area pricing layout descriptions

Item No	Element	Status	Support
1	chargeObjectName	o	
2	chargeObjectRefPoint	o	
3	networkPoints	o	
4	supportingInformation	o	
5	applicableTimeClasses	o	

B.5.4.24 Cordon pricing layout description / Cordon border segment

Item No	Element	Status	Support
1	cordonEntryLocations	o	
2	cordonExitLocations	o	

B.5.4.25 Cordon entry location description

Item No	Element	Status	Support
1	applicableTimeClasses	o	

B.5.4.26 Cordon exit location description

Item No	Element	Status	Support
1	entryLocation	o	
2	applicableTimeClasses	o	

B.5.4.27 Charge reporting events

Item No	Element	Status	Support
1	absoluteTimeEvents	o	
2	relativeTimeEvents	o	
3	travelledDistanceEvents	o	
4	locationEvents	o	
5	feeLimitEvents	o	
6	chargeReportingEventsAuthenticator	o	

B.5.4.28 Absolute time event

Item No	Element	Status	Support
1	randomDelay	o	

B.5.4.29 Relative time event

Item No	Element	Status	Support
1	randomDelay	o	

B.5.4.30 Location event

Item No	Element	Status	Support
1	chargeObject	o	
2	line	o	
3	area	o	

B.5.4.31 Charge report configuration

Item No	Element	Status	Support
1	cccAttributesContent	o	
2	aggregatedSingleTariffClassSessionContent	o	
3	detectedChargeObjectContent	o	
4	listOfRawUsageDataContent	o	
5	vehicleDescriptionContent	o	
6	chargeReportConfigurationAuthenticator	o	

B.6 PICS proforma for the Back End

B.6.1 Identification of the implementation

B.6.1.1 Identification of PICS

Item No	Question	Response
1	Date of statement (DD/MM/YY)	
2	PICS Serial Number	
3	System Conformance Statement Cross Reference	

B.6.1.2 Identification of the implementation and/or system

Item No	Question	Response
1	Service provider or EFC context name	
2	Service provider and EFC context identifier	
3	Version number	
4	Other information	

B.6.1.3 Identification of the Back End supplier

Item No	Question	Response
1	Organization name	
2	Contact name(s)	
3	Address	
4	Phone number(s)	
5	e-mail address	
6	Other information	

B.6.1.4 Identification of the Back End

Item No	Question	Response
1	Brand name	
2	Type, version	
3	Manufacturer ID	
4	Equipment class	
5	Serial number(s)	
6	Other information	

B.6.2 Identification of the standard

Item No	Question	Response
1	Title, Reference No, publication date of the Technical Specification	
2	Technical Specification version	
3	Implemented addenda	
4	Implementer's guide version number	
5	Implementation defect reports	
6	Other information	

B.6.3 Global statement of conformance

Are all mandatory capabilities implemented? (Yes/No) ……

NOTE 1 Answering “No” to this question indicates non-conformance to the specification. Non-supported mandatory capabilities are to be identified in the PICS, with an explanation of why the implementation is non-conforming, on pages attached to the PICS proforma.

B.6.4 PICS proforma tables

This part of the PICS proforma identifies the supported application context, the communication services and attributes (ADUs).

B.6.4.1 ADU and ADU header

Item No	Element	Status	Support
1	Iso17575-3Adu	m	
2	AduAuthenticator	o	

B.6.4.2 Communication services support

Item No	Element	Status	Support
1	Iso17575-2 Communication services supported	m	

B.6.4.3 EFC Attributes

Item No	Element	Status	Support
1	TollContextOverview	m	
2	TariffTable	m	
3	TariffClassDefinition	m	
4	LocalVehicleClassDefinition	m	
5	TimeClassDefinition	o	
6	UserClassDefinition	o	
7	TollContextLayout	m	
8	ChargeReportingEvents	m	
9	ChargeReportConfiguration	m	

B.6.4.4 Toll context overview

Item No	Element	Status	Support
1	tollSchemeName	o	
2	tollContextBoundingBoxes	o	
3	TollContextOverviewAuthenticator	o	

B.6.4.5 Toll scheme types

Item No	TollSchemeType	Status	Support
1	roadSectionPricing	o	
2	areaPricingDistance	o	
3	areaPricingTime	o	
4	cordonPricing	o	

B.6.4.6 Operational status

Item No	TollSchemeType	Status	Support
1	stopsOperationAt	o	

B.6.4.7 Tariff table and tariffs

Item No	Element	Status	Support
1	vat	o	
2	intervalScaleParameters	o	
3	offsetFee	o	
4	minFee	o	
5	thresholdFee	o	
6	maxFee	o	
7	roundingRules	o	
8	tariffTableAuthenticator	o	

B.6.4.8 Tariff class definitions and tariff classes

Item No	Element	Status	Support
1	localVehicleClasses	m	
2	timeClasses	o	
3	locationClasses	o	
4	userClasses	o	
5	tariffClassDefinitionAuthenticator	o	

B.6.4.9 Local vehicle class definitions and local vehicle classes

Item No	Element	Status	Support
1	ordinalElements	o	
2	priorityValue	o	
3	localVehicleClassDefinitionAuthenticator	o	

B.6.4.10 Nominal vehicle parameters

Item No	Element	Status	Support
1	vehicleClasses	m	
2	vehicleAxlesNumber	o	
3	euroValue	o	

B.6.4.11 Ordinal vehicle parameters

Item No	Element	Status	Support
1	vehicleLengthOverall	o	
2	vehicleHeightOverall	o	
3	vehicleWidthOverall	o	
4	vehicleFirstAxlesHeight	o	
5	vehicleAxlesNumber	o	
6	vehicleMaxLadenWeight	o	
7	vehicleTrainMaximumWeight	o	
8	vehicleWeightUnladen	o	
9	vehicleWeightLaden	o	
10	euroValue	o	
11	copValue	o	
12	vehicleClass	o	
13	co2EmissionValue	o	

B.6.4.12 Time class definitions and time classes

Item No	Element	Status	Support
1	nominalElements	o	
2	ordinalElements	o	
3	priorityValue	o	
4	timeClassDefinitionAuthenticator	o	

B.6.4.13 Nominal time class parameters

Item No	Element	Status	Support
1	weekdays	o	
2	dates	o	
3	classesSetExternally	o	

B.6.4.14 Ordinal time class parameters

Item No	Element	Status	Support
1	weekdays	o	
2	absoluteTimeOfDay	o	
3	relativeTimePeriods	o	
4	periodsInYear	o	

B.6.4.15 User class definitions and user classes

Item No	Element	Status	Support
1	contractTypes	o	
2	actualNumberOfPassengers	o	
3	userClassDefinitionAuthenticator	o	

B.6.4.16 Toll context layout

Item No	Element	Status	Support
1	geoRefPoint	o	
2	tollContextAuthenticator	o	

B.6.4.17 Supported context types

Item No	Element	Status	Support
1	sectionPricingLayout	o	
2	areaPricingLayout	o	
3	cordonPricingLayout	o	

B.6.4.18 Section pricing layout description

Item No	Element	Status	Support
1	chargeObjectName	o	
2	chargeObjectRefPoint	o	
3	networkPoints	o	
4	pathStructureTowards	o	
5	pathStructureOnwards	o	
6	supportingInformation	o	
7	realDistance	o	
8	applicableTimeClasses	o	
9	storageRequired	o	

B.6.4.19 Point

Item No	Element	Status	Support
1	pointIdentifier	o	
2	absolutePointCoordinates	o	
3	relativePointCoordinates	o	

B.6.4.20 Link

Item No	Element	Status	Support
1	intermediatePoints	o	

B.6.4.21 Supporting Point

Item No	Element	Status	Support
1	distanceToNextRoad	o	

B.6.4.22 Area pricing layout description

Item No	Element	Status	Support
1	roadNetworks	o	

B.6.4.23 Road network objects in area pricing layout descriptions

Item No	Element	Status	Support
1	chargeObjectName	o	
2	chargeObjectRefPoint	o	
3	networkPoints	o	
4	supportingInformation	o	
5	applicableTimeClasses	o	

B.6.4.24 Cordon pricing layout description / Cordon border segment

Item No	Element	Status	Support
1	cordonEntryLocations	o	
2	cordonExitLocations	o	

B.6.4.25 Cordon entry location description

Item No	Element	Status	Support
1	applicableTimeClasses	o	

B.6.4.26 Cordon exit location description

Item No	Element	Status	Support
1	entryLocation	o	
2	applicableTimeClasses	o	

B.6.4.27 Charge reporting events

Item No	Element	Status	Support
1	absoluteTimeEvents	o	
2	relativeTimeEvents	o	
3	travelledDistanceEvents	o	
4	locationEvents	o	
5	feeLimitEvents	o	
6	chargeReportingEventsAuthenticator	o	

B.6.4.28 Absolute time event

Item No	Element	Status	Support
1	randomDelay	o	

B.6.4.29 Relative time event

Item No	Element	Status	Support
1	randomDelay	o	

B.6.4.30 Location event

Item No	Element	Status	Support
1	chargeObject	o	
2	line	o	
3	area	o	

B.6.4.31 Charge report configuration

Item No	Element	Status	Support
1	cccAttributesContent	o	
2	aggregatedSingleTariffClassSessionContent	o	
3	detectedChargeObjectContent	o	
4	listOfRawUsageDataContent	o	
5	vehicleDescriptionContent	o	
6	chargeReportConfigurationAuthenticator	o	

Annex C
(informative)
How to use context data defining the properties of an EFC regime

C.1 General

The following guidelines illustrate how EFC context data may be used defining the properties of a specific EFC regime. The intention of this annex is supporting the understanding of the basic principles of this part of ISO/TS 17575.

However, in order to reduce complexity, some of the options available in this part of ISO/TS 17575 are not mentioned here if their use is not of a high importance for the general understanding. So, for use in real operational life, the normative part of this part of ISO/TS 17575 shall be used instead.

The task defining the regime properties is allocated to the Back End operator. Here the context data are provided supporting all the Front Ends the service provider is responsible for.

In cases of split responsibilities of the roles of the toll charger and the service provider the service provider may take the scheme definition from similar definitions distributed by the appropriate toll chargers.

C.2 The evaluation process determining the basic fee

The process illustrated in Figure C.1 determines the basic fee which shall be applied per charge objects used. This basic fee shall be used to multiply the assembled charge relevant parameter. This is a price either per event (in cordon pricing schemes) or per time (in time related area pricing schemes) or per km (for distance based tariffs like sectioned road tolling or distance based area pricing).

The process starts determining the actual applicable vehicle class. Therefore the specific vehicle nominal and ordinal classification parameters shall be compared with the values provided in the EFC context data in the list of the elements `LocalVehicelClass`. A vehicle class is recognised as being valid if all nominal scaling parameters provided in the data element `localVehicleClass` are equal with the actual specific vehicle parameters and all ordinal parameter limits provided in the `localVehicleClass` corresponds with the actual specific vehicle parameters.

The specific vehicle parameters are provided by the Front End configuration parameters or the variable parameters were set by the User. If more than one definition of local vehicle classes is fulfilled then the data element `priorityLevel` can be used selecting the vehicle class with the highest priority.

The next parameter which needs to be fixed when addressing the basic fee table is the actual contract class. Normally this should be a stable identifier valid for a certain contract of the user. However it may change if vehicles are used for different purposes which might have an influence of the tariff to be applied. In this case the Front End must provide means selecting one of the possible contract classes.

The charge object specific location class is an identifier provided in the EFC context data in the data element `TollContextLayout`. This was set in a way that the corresponding charge object is allocated to the price class of this certain location. This may be used distinguishing between high price and low price locations like urban and rural locations where the tariff of a comparable usage differs.

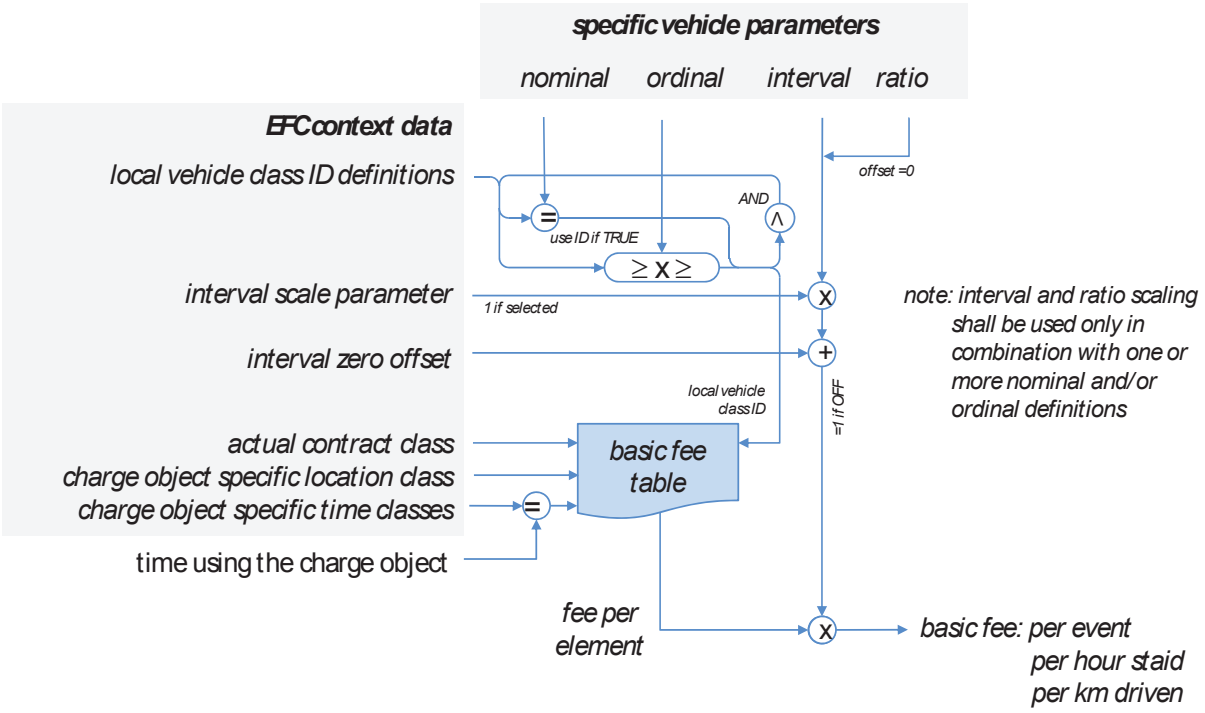


Figure C.1 — Mapping of EFC context data and specific vehicle parameters determining the basic fee

The actual time class is the last tariff modifier and shall be determined according to the rules illustrated in clause C.4. This may provide more than one actually valid time classes. This may happen if e.g. one time class is the week end, another time class is a certain date in the year and another time class covers a certain time span of a day. All these classes can be valid at the same time. And in addition to that the definition **`applicableTimeClasse`** shall be used eliminating time classes not to be used for certain charge objects. If still more than one time class remains then the parameter **`priorityValue`** in **`timeClasses`** shall be used to select the time class with the highest priority.

If an external time class is set by external means then this shall be used in the same way as the time when evaluating the nominal scale parameters.

The access to the basic fee table should be understood as a two step process. The first step is determining the **`tariffClass`** by comparing the group of 4 identifiers for the tariff modifiers vehicle class, contract class, location class and the time class with the **`tariffClassId`** provided in the **`tariffClass`** data element. In the second step this **`tariffClass`** shall be used to read from the **`tariffs`** data element the corresponding data elements **`basicFeePerChargeUnit`** and **`intervalScaleParameter`**.

If no interval scale parameter is defined for this tariff class then the corresponding value shall be set to 1 (the later multiplication does not change the value). If an interval scale parameter is defined then the corresponding value from the specific vehicle parameters shall be selected. This value may be modified according to optional parameters **`zeroOffset`**, **`resolution`** and **`max`**.

Now the two values evaluated, the **`basicFeePerChargeUnit`** and the eventually modified value of the **`intervalScaleParameter`** shall be multiplied. The output is the basic fee to be applied for this charge object for each unit of the charge relevant value (distance, time or events).

C.3 The definition of time classes

Time classes are specified in the EFC context data by setting parameters in the `timeClasses` data element. A definition of a single time class consists of a `timeClassId` which may be used to address this specific time class, it contains the `priorityLevel` allowing to place this time class within the ranking relative to others, it consist finally of the nominal and ordinal scaled parameters to be compared with the time. The time shall be understood as the time a charge object is or was used. The time used shall be the local time according to the `timeZone` defined in the `TollContextOverview`. An optional local daylight saving time period shall be applied.

As nominal scaling parameters the weekdays, months and discrete dates of the year are applicable. Also external set time classes are processed as nominal parameters.

Ordinal parameters are a time span within a day, a relative time period relative to the time the vehicle has entered a charge object, a contiguous group of weekdays and a contiguous group of days in a year.

C.4 The time class evaluation algorithm

To determine if a specified time class is valid for a specific time, this time shall be used evaluating some specific parameters.

The first is the time of the day modulo 24 hours. Then the weekday needs to be calculated. The next parameter required is the date of the year. The last one is the time duration from entering the charge object until the time analysed.

These parameters shall be used comparing for each defined time class the provided nominal values as well as the ordinal limits of the field of validity with the analysed values of the relevant reference time. If all the nominal parameters are equivalent to the actual ones and all ordinal parameters cover the respective actual values then this time class is or was valid at the reference time.

With this more than one on the define time classes can be valid.

A graphical presentation of this process is illustrated in Figure C.2.

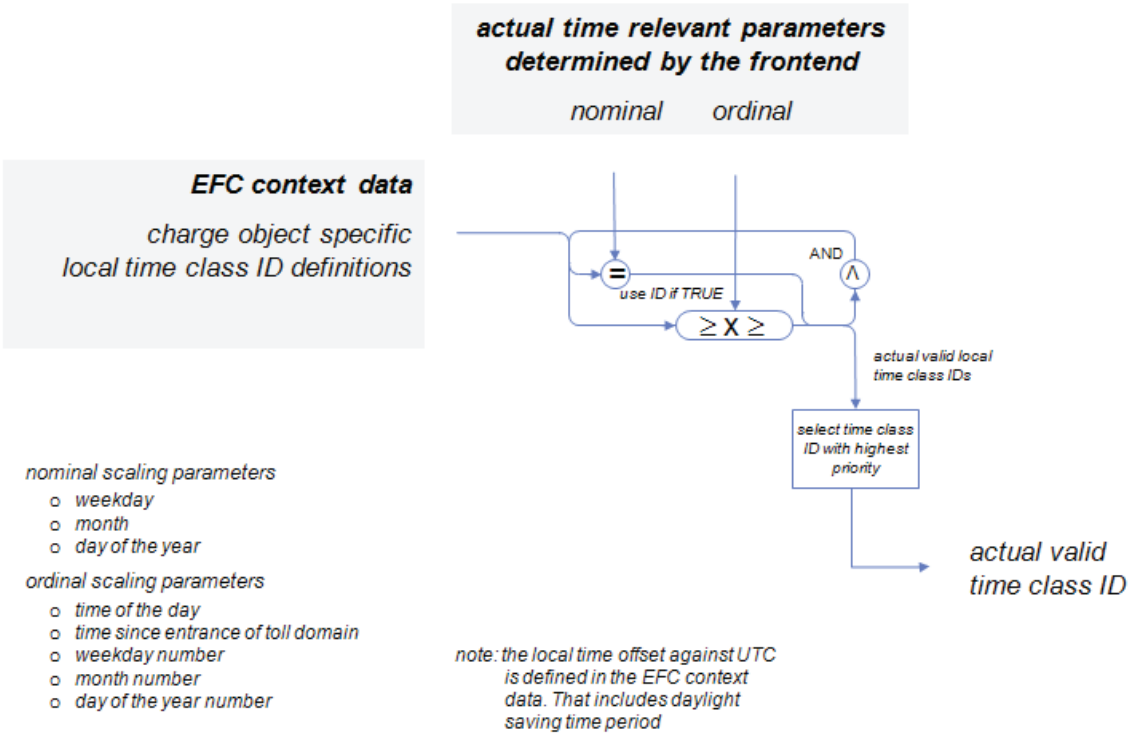


Figure C.2 — Mapping of EFC context data and Front End specific timing parameters determining the actual valid time class

C.5 Example of a charge object recognition algorithm for sectioned roads

It is outside the scope of this part of ISO/TS 17575 to provide the charge object recognition algorithm for sectioned roads as for other types of charge objects. However, understanding the `sectionLayout` data element of the EFC context data requires having at least an example of an evaluation algorithm in mind.

The example presented in Figure C.3 is split into two basic evaluation steps. The first step evaluates if the orthogonal distance from any of the known trajectories towards the charge point may allow assuming the vehicle has used this trajectory and a second step evaluates the heading deviation in the same way.

Both of these steps are again split into two phases. The first phase evaluates if all the single measurement samples measured in the full path towards the charge point are within a maximum tolerance of measurement. A second phase estimates if the mean square deviation between the measured value to the known value is below a tolerance level.

This provides 4 Boolean expressions. If all these 4 Boolean results are TRUE then the output of this evaluation algorithm is TRUE meaning that the vehicle might have used this charge object. This decision has an error probability depending on the measurement accuracy and noise, on the accuracy and noise of the provided parameters in the `Layout` and on the tolerances used.

Note: It is of utmost importance that the provider of the context data and the manufacture of the Front End agree on the values of the tolerances.

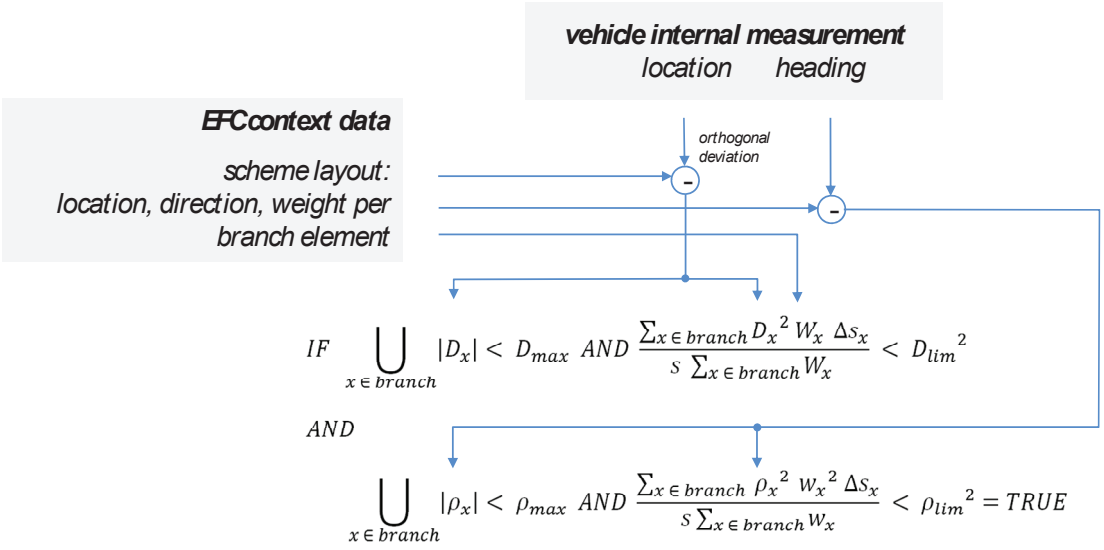


Figure C.3 — The charge object recognition algorithm

where D = orthogonal location deviation (max and lim device specific)
ρ = heading deviation (max and lim device specific)
W = distance weight factor per branch element
w = heading weight factor per branch element

Note: Measurement x is element of branch if it is element of the road course from the leaf to the root.

The Boolean equation in Figure C.3 shall be read as:

If all absolute orthogonal (to the direction of the road trajectory) measured location deviations D_x which are element of the branch (towards the charge point) are less than the tolerance D_{max} and

the sum of all squared orthogonal measured location deviations weighted by the squared weight factor W_x (derived from the optionally provided **`distanceToNextRoads`** in the **`layout`**) and weighted by the incremental travelled distance Δs_x relative to the branch length s and the sum of all weight factors are less than the squared tolerance D_{lim} and

if all absolute measured heading deviations (to the direction of the road trajectory) ρ_x which are element of the branch (towards the charge point) are less than the tolerance ρ_{max} and

the sum of all squared measured heading deviations weighted by the squared weight factor w_x (optionally provided in the **`layout`**) and weighted by the incremental travelled distance Δs_x relative to the branch length s and the sum of all weight factors are less than the squared tolerance ρ_{lim} then the vehicle has used this section of the toll road (the charge object).

However, any other evaluation algorithm providing the same result with a similar error probability may also be used.

Annex D
(informative)
Examples using EFC context data for scheme definitions

D.1 General

The following examples illustrate how EFC context data as defined in this part of ISO/TS 17575 may be used when defining existing EFC schemes. It is based on the scheme definitions available in 2008 and may be adapted whenever required. These examples are not complete. Scheme identifiers and other obvious and self explaining elements were left out for better readability. Some of the listed data elements at higher levels are not explained in details either if they are obvious or if they define geographical details.

These examples shall be used to help understand the overall concepts and to gather ideas for coding a scheme. They shall not be copied and used for operational use.

D.2 Example for a section tolling scheme

D.2.1 Introduction

The following example explains how this part of ISO/TS 17575 can be used to define the context description in a simple section tolling scheme for trucks.

The example below does not contain a complete set of attributes and data elements required to implement a full section pricing context. It gives a choice to explain how data elements can be used.

D.2.2 Rules of the EFC scheme

Liable Vehicle Classes

All trucks 3.5 tonnes and above. Fee depends on number of axles and the engine pollution class. (Table D.1)

Table D.1 — Vehicle class definitions

Axles number	Emission Class	Resulting Vehicle Class
2 and 3	Euro 1 and Euro 2	A
	Euro 3	B
	Euro 4	C
	Euro 5 and better	D
4 and more	Euro 1 and Euro 2	E
	Euro 3	F
	Euro 4	G
	Euro 5 and better	H

Time dependency

Charging at any time. No time dependent fees are defined.

Charged road network

All defined charge objects (road section pricing principle)

Tariff definitions

The tariffs shall be applied according to the Table D.2. In addition to these distance based fee a minimum fee of 1,50 EUR shall apply.

Table D.2 — Tariff table

Vehicle Class	Tariff [EUR/km]
A	0,32
B	0,25
C	0,18
D	0,11
E	0,35
F	0,28
G	0,21
H	0,15

D.2.3 Coding of data elements

Table D.3 — Attribute TollContextOverview

Data element	Content	Remark
tollCharger	Truck toll SA registered in Example Country	Coding includes country identifier and an identifier for the toll charger within this country
tollContext	Truck toll scheme A applicable in Example Country	Coding includes country identifier and an identifier for the toll context within this country
tollSchemeName	“Example Truck Toll”	Optional and present
operationalStatus		
startsOperationAt	01.01.2010	
stopsOperationAt	*- n.a. -*	optional and not present
timeZone	MET	To be coded as difference in minutes compared to UTC
tollContextBoundingBoxes	List of boxes covering the charged road network	
tollContextOverviewVersion	Version 1.23, valid from 01.03.2011	
tollContextOverviewAuthenticator	Authenticator	Optional and present

Table D.4 — Attribute TariffTable

Data element	Content	Remark
tariffs	Refer to Table D.5	
roundingRule	Always rounding up	optional and present
tariffTableVersion	Version 1.19 valid from 01.09.2010	
tariffTableAuthenticator	Authenticator	Optional and present

Table D.5 — Data element Tariff

Data element	Content	Remark
Tariff T1		For tariff T1
tariffClassId	1	
chargeUnit	100 m	
currency	EUR	
basicFeePerChargeUnit	0,32	= 0,32 EUR
vat	- n.a. -	optional and not present
intervalScaleParameter	- n.a. -	optional and not present
offsetFee	- n.a. -	optional and not present
minFee	15000	= 1,50 EUR (15.000 x 0,0001 EUR)
thresholdFee	- n.a. -	optional and not present
maxFee	- n.a. -	optional and not present
Tariff T2		For tariff T2
tariffClassId	2	
chargeUnit	100 m	
currency	EUR	
basicFeePerChargeUnit	0,25	= 0,25 EUR
...	...	see above at tariff T1
Tariff T3		For tariff T3
...	...	see above at tariff T1
basicFeePerChargeUnit	0,1800	= 0,18 EUR
...	...	

Table D.6 — Attribute LocalVehicleClassDefinition

Data element	Content	Remark
localVehicleClasses	Refer to Table D.7	
tariffTableVersion	Version 1.03 valid from 01.03.2010	
tariffTableAuthenticator	Authenticator	Optional and present

Table D.7 — Data element localVehicleClasses

Data element	Content	Remark
LocalVehicleClass A		For vehicle class A
localVehicleClassId	1	
nominalElements		
vehicleClasses	Trucks above 3.5t	Coding according to EN 15509
vehicleAxlesNumber	2, 3	Coding according to ISO 14906
euroValues	Euro 0, 1, 2	Coding according to ISO 14906
classesSetExternally	- n.a. -	optional and not present
ordinalElements	- n.a. -	optional and not present
priorityValue	250	
LocalVehicleClass B		For vehicle class B
localVehicleClassId	2	
nominalElements		
vehicleClasses	Trucks above 3.5t	Coding according to EN 15509
vehicleAxlesNumber	2, 3	Coding according to ISO 14906
euroValues	Euro 3	Coding according to ISO 14906
...	...	See above at vehicle class A
LocalVehicleClass C		For vehicle class C
localVehicleClassId	3	
nominalElements		
vehicleClasses	Trucks above 3.5t	Coding according to EN 15509
vehicleAxlesNumber	2, 3	Coding according to ISO 14906
euroValues	Euro 4	Coding according to ISO 14906
...	...	See above at vehicle class A

Bibliography

[1] ISO 6709, *Standard representation of geographic point location by coordinates*

[2] ISO/IEC 9646-7, *Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 7: Implementation Conformance Statements*

[3] ISO/TS 17575-4, *Electronic fee collection — Application interface definition for autonomous systems — Part 4: Roaming*

[4] 2009/750/EC, *Commission Decision of 6 October 2009 of the European Electronic Toll Service and its technical elements*

[5] DIN 1333, *Presentation of numerical data*

Technical Corrigendum 1 to ISO/TS 17575-3:2011 was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*, in collaboration with Technical Committee CEN/TC 278, *Road transport and traffic telematics*.

Page 11, 8.3.2, second paragraph and second sentence below Table 3

Replace

'The data type shall be **`ContextId,`** defined in ISO/TS 17575-1:2010.'

with

'The data type shall be **`ContextId,`** as defined in A.2.'

Page 12, 8.3.2

Replace

'− data element **`southernLatitude`** shall contain the lower (southernmost) latitude value (data type **`Latitude)`**,

− data element **`northernLatitude`** shall contain the larger (northernmost) longitude value (data type **`Latitude)`**,

− data element **`westernLongitude`** shall contain the lower (westernmost) latitude value (data type **`Longitude)`** and

− data element **`easternLongitude`** shall contain the larger (easternmost) latitude value (data type **`Longitude)`**.'

with

'− data element **`southernLatitude`** shall contain the lower (southernmost) latitude value (data type **`Latitude)`**,

− data element **`northernLatitude`** shall contain the larger (northernmost) latitude value (data type **`Latitude)`**,

− data element **`westernLongitude`** shall contain the lower (westernmost) longitude value (data type **`Longitude)`** and

− data element **`easternLongitude`** shall contain the larger (easternmost) longitude value (data type **`Longitude)`**.'

Page 37, 8.3.4.3.1, Table 19, second column, row 4

Replace

'**`cordonEntryData'`**

with

`'cordonEntryLocation'`

Page 37, 8.3.4.3.1, Table 19, second column, row 5

Replace

`'cordonExitData'`

with

`'cordonExitLocation'`

Page 37, 8.3.4.3.1, third paragraph below Table 19

Replace

'**`cordonEntryData'`**

with

```
'cordonEntryLocation'
```

Page 51, Annex A

Replace

```
'FROM CccModule {iso standard 12813 modules(0) efc(0) version(1)};'
```

with

```
FROM CccModule {iso standard 12813 modules(0) ccc(0) version(1)};'.
```

Page 51, A.2

Replace

```
'VehicleWeightLimits, EnvironmentalCharacteristics,'
```

with

```
'VehicleWeightLimits, EnvironmentalCharacteristics, EuroValue,'
```

i.e. importing the EuroValue ASN.1 type from ISO 14906 instead of copying its definition.

Page 55, A.2

Delete

```
'EuroValue ::= ENUMERATED { --same definition as in ISO 14906
    noEntry        (0),
    euro-1         (1),
    euro-2         (2),
    euro-3         (3),
    euro-4         (4),
    euro-5         (5),
    euro-6         (6),
    reservedForUse1 (7)
    }'
```

Page 59, A.2

Replace

```
      'southernLongitude        Longitude,
      northernLongitude         Longitude,
      westernLatitude           Latitude,
      easternLatitude           Latitude'
```

with

```
      'southernLatitude         Latitude,
      northernLatitude          Latitude,
      westernLongitude          Longitude,
      easternLongitude          Longitude'
```

Bureau of Indian Standards

BIS is a statutory institution established under the *Bureau of Indian Standards Act*, 1986 to promote harmonious development of the activities of standardization, marking and quality certification of goods and attending to connected matters in the country.

Review of Indian Standards

Amendments are issued to standards as the need arises on the basis of comments. Standards are also reviewed periodically; a standard along with amendments is reaffirmed when such review indicates that no changes are needed; if the review indicates that changes are needed, it is taken up for revision. Users of Indian Standards should ascertain that they are in possession of the latest amendments or edition by referring to the latest issue of 'BIS Catalogue' and 'Standards : Monthly Additions'.

This Indian Standard has been developed from Doc No.: TED 28 (0981).

Amendments Issued Since Publication

Amend No.	Date of Issue	Text Affected

BUREAU OF INDIAN STANDARDS

Headquarters:

Manak Bhavan, 9 Bahadur Shah Zafar Marg, New Delhi 110002
Telephones : 2323 0131, 2323 3375, 2323 9402 *Website*: www.bis.org.in

Regional Offices:		*Telephones*
Central	: Manak Bhavan, 9 Bahadur Shah Zafar Marg, NEW DELHI 110002	2323 7617, 2323 3841
Eastern	: 1/14 C.I.T. Scheme VII M, V. I. P. Road, Kankurgachi, KOLKATA 700054	2337 8499, 2337 8561, 2337 8626, 2337 9120
Northern	: SCO 335-336, Sector 34-A, CHANDIGARH 160022	260 3843, 260 9285
Southern	: C.I.T. Campus, IV Cross Road, CHENNAI 600113	2254 1216, 2254 1442, 2254 2519, 2254 2315
Western	: Manakalaya, E9 MIDC, Marol, Andheri (East), MUMBAI 400093	2832 9295, 2832 7858, 2832 7891, 2832 7892

Branches: AHMEDABAD. BENGALURU. BHOPAL. BHUBANESHWAR. COIMBATORE. DEHRADUN. FARIDABAD. GHAZIABAD. GUWAHATI. HYDERABAD. JAIPUR. KOCHI. LUCKNOW. NAGPUR. PARWANOO. PATNA. PUNE. RAJKOT. VISAKHAPATNAM.

Published by BIS, New Delhi