BS EN 1545-1:2015

BSI Standards Publication

Identification card systems — Surface transport applications

Part 1: Elementary data types, general code lists and general data elements

bsi.

...making excellence a habit.™

National foreword

This British Standard is the UK implementation of EN 1545-1:2015. It supersedes BS EN 1545-1:2005 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/17, Cards and personal identification.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Published by BSI Standards Limited 2015

ISBN 978 0 580 80849 4

ICS 35.240.15

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 April 2015.

Amendments/corrigenda issued since publication

Date	Text affected

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN 1545-1

April 2015

ICS 35.240.15

Supersedes EN 1545-1:2005

English Version

Identification card systems - Surface transport applications - Part 1: Elementary data types, general code lists and general data elements

Systèmes de cartes d'identification - Applications pour le transport terrestre - Partie 1 : Types de données élémentaires, codes généraux et éléments de données généraux

Identifikationskartensysteme - Landgebundene Transportanwendungen - Teil 1: Elementare Datentypen, allgemeine Codelisten und generelle Datenelemente

This European Standard was approved by CEN on 27 September 2014.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and United Kingdom.

EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

Contents

Page

Foreword

This document (EN 1545-1:2015) has been prepared by Technical Committee CEN/TC 224 "Personal identification, electronic signature and cards and their related systems and operations", the secretariat of which is held by AFNOR.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by October 2015 and conflicting national standards shall be withdrawn at the latest by October 2015.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN [and/or CENELEC] shall not be held responsible for identifying any or all such patent rights.

This document supersedes EN 1545-1:2005.

This European Standard comprises the following parts, under the general title "Identification card systems - Surface transport applications":

- General part:

Part 1: Elementary data types, general code lists and general data elements;

- Sector specific part:

Part 2: Transport and travel payment related data elements and codelists.

According to the CEN-CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

Introduction

ICs offer far greater opportunities for use in surface transport applications (STA) when compared to magnetic stripe and barcoded cards. The standardisation of data elements, which is the purpose of this European Standard, facilitates the use of ICs across multiple transport applications and operators, and in a variety of transport related terminals. This European Standard also permits application builders to minimise data duplication.

This European Standard contains definitions of data formats, data elements, data types and specifies data elements with associated codelists. It is for use in the creation of surface transport related data structures that may reside on a transport application. Abstract Syntax Notation One (ASN.1) has been used in the definition of data types in this European Standard.

This European Standard provides a comprehensive toolbox of data elements and types as the basis for the creation of data structures to be used in STAs. This European Standard alone does not ensure interoperability; this is left to the application builders. The definition of data structures to be used in STAs is left to applications.

This European Standard has a hierarchical approach:

1. basis for all definitions used in this European Standard is ASN.1 (ISO/IEC 8824);

2. EN 1545-1 standardises its general elements, data types and data elements with associated code lists in accordance with ASN.1;

3. sectoral parts of this European Standard (EN 1545-2) define the sector specific elements and codes. Apart from the sector specific codes that are directly based on ASN.1 all definitions of sector specific data elements have to be based on EN 1545-1 definitions;

4. it is left to applications to define the relevant data structures (data objects) strictly based on the definitions of EN 1545.

4. Any transport application

data structures (objects)

sector specific data elements from EN 1545-sectoral

sector specific codes from EN 1545-sectoral

general data elements from EN 1545-1

elementary data types from EN 1545-1

general data elements with code lists from EN 1545-1

3. EN 1545-sectoral

sector specific data elements

general data elements from EN 1545-1

elementary data types from EN 1545-1

sector specific code lists

codes expressed in ASN.1

2. EN 1545-1

general data elements

elementary data types from EN 1545-1

universal ASN.1 types from ISO/IEC 8824

general data elements with associated code lists

codes expressed in ASN.1

elementary data types

universal ASN.1 types from ISO 8824

1. ISO 8824

universal ASN.1 data types

This European Standard refers to existing ASN.1 encoding rules (transfer syntaxes), such as the basic and packed encoding rules, for use within surface transport applications. However this European Standard does not exclude the use of other encoding rules.

The ASN.1 basic encoding rules (BER) includes significant redundancy in order to make transferred data fully self-defining, which may result in data structures too large to be used in applications on ICs with restricted data storage capacity. Therefore this European Standard allows the use of alternative encoding rules such as the ones based upon the ASN.1 packed encoding rules (PER) (see Clause 9).

The mechanism for how to establish the application context, including the decision as to which encoding rules to use, is outside the scope of this European Standard.

This European Standard does not pretend to identify and specify every possible ASN.1 type that may be used in future applications by application builders. In addition, local systems may be defined in their own way.

This European Standard will be updated and added to over time as new surface transport applications are created, in accordance with the normal CEN practice.

1 Scope

This European Standard specifies data formats, data elements, data types and data elements with associated codelists for general use within surface transport applications (STAs) on ICs.

The mechanism for how to establish the application context, including the decision of which encoding rules to use, is outside the scope of this European Standard.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

EN 1332-4, *Identification card systems - Man-machine interface - Part 4: Coding of user requirements for people with special needs*

EN ISO 3166-1, *Codes for the representation of names of countries and their subdivisions - Part 1: Country codes (ISO 3166-1)*

ISO 639-2, *Codes for the representation of names of languages — Part 2: Alpha-3 code*

ISO 4217, *Codes for the representation of currencies*

ISO/IEC 5218, *Information technology — Codes for the representation of human sexes*

ISO/IEC 7816-5:2004, *Identification cards — Integrated circuit cards — Part 5: Registration of application providers*

ISO/IEC 7816-6:2004, *Identification cards — Integrated circuit cards — Part 6: Interindustry data elements for interchange*

ISO/IEC 8824-1:2008, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation — Part 1*

ISO/IEC 8825-1:2008, *Information technology — ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER) — Part 1*

ISO/IEC 8825-2:2008, *Information technology — ASN.1 encoding rules: Specification of Packed Encoding Rules (PER) — Part 2*

ISO 14816, *Road transport and traffic telematics — Automatic vehicle and equipment identification — Numbering and data structure*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
account
a precise list or enumeration of financial transactions held in a central location, used for payment for services. When payment is made through the use of a card, the card identifies the centrally held account

3.2
application
file structure, directory entries and security scheme loaded in part onto a smart card to perform a particular set of functions. A card may support one or more applications

3.3
basic data element
data element with a single value, defined with reference to a single data type

3.4
code
data element, with an associated mandatory codelist, expressed in ASN.1 types, namely:

- enumerated type;

- integer type

3.5
constructed data element; composite data element
group of basic data elements

3.6
data element
single store for an irreducible datum value

3.7
elementary data type
data element is an elementary data type when it occurs in more than one data construct whilst keeping the same semantic and syntactic values in each occurrence

An elementary data type is either:

- data type defined with reference to a single universal data type,

- or an enumeration

3.8
general data element
data element defined with reference to universal or elementary data type

3.9
journey
complete sequence of one or more journey legs required to achieve a specific purpose at a specific destination. This sequence may include the use of more than one vehicle and using more than one transport mode

3.10
network
all coordinated lines of road based, rail based, water based transport within a defined geographical area and/or under a specified authority where a card is uniquely used

3.11
sector specific data element
data element defined with reference to a general data element or an elementary data type

3.12
universal data type
universal ASN.1 Type defined in ISO/IEC 8824-1:2008

Note 1 to entry: UTF-8 strings are used to represent characters. Users should note that each character may occupy 1 to 4 bytes, depending upon the character. Length of such strings is therefore a count of characters, not a count of bytes.

4 Abbreviations

For the purposes of this document, the following abbreviations apply.

AID	Application Identifier [ISO/IEC 7816-5:2004]
ASN.1	Abstract Syntax Notation One [ISO/IEC 8824-1:2002]
BCD	Binary Coded Decimal
BER	Basic Encoding Rules [ISO/IEC 8825-1:2008]
BIBO	Be-in, Be-out
CAD	Card accepting device
CICO	Check-in, Check-out
CIBO	Check-in, Be-out
EAN	European Article Numbering
GDF	Geographical Data File
GSM	Global System for Mobile Communication
ICC	Integrated Circuit Card
IEP	Intersector Electronic Purse licensed according to the EC e-Money Directive 2000/46
MII	Major Industry Identifier [ISO/IEC 7812-1:2000]
NOTOK	Not OK
PER	Packed Encoding Rules [ISO/IEC 8825-2:2008]
RFU	Reserved for Future Use
STA	Surface Transport Application
STR	Stored Travel Rights
VAT	Value Added Tax
WIWO	Walk-in, Walk-out

5 Approach for definition of data types and data elements

5.1 Data types and data elements

This European Standard uses Abstract Syntax Notation One (ASN.1) to define data types and data elements. This enables simple and structured data to be defined without implying any specific transfer syntax (encoding rules), which will be application - and environment dependant.

Basic data elements are always defined with reference to a data type.

If a group of basic data elements are always used together we define this group as a composite data element. For composite data elements, there is no data type definition.

Decisions on transfer syntax are left to applications.

5.2 ASN.1 type naming conventions

ASN.1 type naming conventions shall be done in accordance with ISO/IEC 8824-1. This implies that:

- where possible, the meaning of the data type or data element is implied through the names being selected;

- where a data type or data element is a composition of other data types or data elements, the data type name or data element name is still a single sequence of alphabetical characters commencing with a capital letter, however capitals are used within the name to impart the corresponding meaning.

In general, data type names or data element names are related to the name of the data types or data elements from which they are constructed, the application or application area name and the function related to the data.

5.3 Existing standards

If an ASN.1 type is already defined as part of another European Standard and if it is relevant for usage within STAs, then this ASN.1 type will be defined in this European Standard for use within the surface transport domain. However this European Standard refers to ISO/IEC 7816-6 to enable STA to use the data types and data elements of ISO/IEC 7816-6.

5.4 Value range identifiers

To enable several types of encoding rules, some ASN.1 types in this European Standard are constrained by value range identifiers. The value range identifiers are defined in European Standards using EN 1545.

5.5 Size constraints

To save memory space within an application the size of any data element defined in this European Standard may be constraint according to ASN.1 rules.

6 Elementary data types

6.1 Address

A string of characters representing a town or an address with a street, a house number, a town.

Address ::= UTF8String (SIZE(0..255))

6.2 Amount

The value of a monetary transaction. Note that an amount is always a positive integer or zero. Amount always includes VAT where applicable.

Amount {INTEGER:amountRange} ::= INTEGER (0..amountRange)

If not otherwise specified by means of another data element, the currency units applicable to Amount shall be 0,01 Euro.

Value Assignment: network specific

6.3 ApplicationInstanceNumber

A unique identification number for an application instance

ApplicationInstanceNumber ::= ReferenceNumber

6.4 Authenticator

A cryptographic transformation of a data set that allows a recipient to prove the integrity of the data set and/or the identity of its source. The algorithm and the data being protected by this data element is implicitly given by the context, determined by the application, in which the element is used.

Typical implementations are Message Authentication Codes (MACs) or digital signatures. Authenticator ::= OCTET STRING (SIZE(1..authenticatorSize))

Value Assignment : Application specific

6.5 BCDStringType

This type is retained only to support definitions as defined in the referenced ISO standards and shall not be used in new data element definitions.

BCDStringType is applied for Binary Code Decimal (BCD) representation. This data type is used to represent one decimal digit in one semi-octet (4 bits). BCDStringType is based on the ISO/IEC 8824-1 ‘CharacterStringType’.

BCDString ::= CHARACTER STRING(WITH COMPONENTS {

identification (WITH COMPONENTS {

fixed PRESENT }) })

BCDStringType ::= BCDString

Value Assignment : BCDStringValue ::= bstring | hstring

When using the “bstring” notation, the leftmost bit shall be the most significant bit of the first octet.

When using the “hstring” notation, the leftmost hexadecimal digit shall be the most significant semi-octet of the first octet.

When an element contains more than one BCD character, it shall be padded with leading zeros where this is necessary.“

Permitted digits are: 0,1, .. 9.

EXAMPLE

Where ASN.1 BER is used as the transfer syntax, the decimal number 578 is represented as ‘0578’H or ‘0000010101111000’B in BCDString. The decimal number 578 is preceded by an ASN.1 identifier (i.e. ASN.1 tag) and a length indicator.

Where ASN.1 PER is used as the transfer syntax, the decimal number 578 is represented or ‘10101111000’B in BCDString.

6.6 BitMap

A data element where each individual bit within the element shall be treated as Boolean and relates to a specific function.

BitMap {INTEGER : bitMapRange } ::= BIT STRING(SIZE (bitMapRange))

6.7 Capacity

The value of volume or occupancy of either a vehicle or a cargo consignment expressed in a unit of capacity. Refer to 6.38 MeasuredParameters for explanation.

Capacity ::= Value {capacity}

6.8 CompanyId

Identification of a company or other organisation within a network.

CompanyId {INTEGER : companyIdSize} ::= OCTET STRING (SIZE(companyIdSize))

6.9 Counter

Represents a value that can be modified during the lifecycle of this element. This element is always positive or zero.

Counter {INTEGER : counterRange} ::= INTEGER (0..counterRange)

6.10 CountryAlpha

The alphabetic reference to a country shall be in accordance with the conventional identification of countries as defined in EN ISO 3166-1, three characters.

CountryAlpha ::= PrintableString (SIZE(3))

6.11 CountryNumeric

The unique designation of a country shall be as defined in EN ISO 3166-1 using the ISO numeric representation.

CountryNumeric ::= INTEGER (0..1023)

6.12 Currency

The unique designation of a currency shall be as defined in ISO 4217 using the ISO numeric representation.

Currency ::= INTEGER (0..1023)

6.13 Databin

Datatype used for defining elements reserved for network specific use.

Databin {INTEGER : databinSize} ::= BIT STRING (SIZE (0..databinSize))

Value Assignment : a NetworkId specific value.

6.14 DateCompact

Formatted date field, where the date is expressed in years, months and days.

```
DateCompact ::= SEQUENCE {
    year        BIT STRING(SIZE(7)),
```

```
    month    BIT STRING(SIZE(4)),
    day      BIT STRING(SIZE(5))
}
```

Value Assignment - Octet Aligned : ‘yyyyyyymmmmddddd’B (16 bits)

‘yyyyyyy’B : The number of years after 1990. (1990 = ‘0000000’B)

‘mmmm’B : The month within the given year. Value Range: ‘0001’B..’1100’B

‘ddddd’B : The day within the given month. Value Range: ‘00001’B..’11111’B

‘0000000000000000’B denotes explicitely no date.

6.15 Datef

Date expressed in a readily printable numeric format.

```
Datef ::= SEQUENCE {
        year     BCDString (SIZE(2)),
        month    BCDString (SIZE(1)),
        day BCDString (SIZE(1))
}
```

Value Assignment:

'00000000'H denotes explicitly no date.

‘yyyy’H = year

‘mm’H = month

‘dd’H = day

6.16 DateStamp

Number of days relative to 1 January 1997, where 1 January 1997 is day 0.

```
DateStamp ::= BIT STRING (SIZE(14))
```

Value Assignment : ‘ddddddddddddd’B (14 bits)

6.17 DateTimeCompact

Concatenation of DateCompact and TimeCompact

```
DateTimeCompact ::= SEQUENCE {
    date     DateCompact,
```

time TimeCompact

}

6.18 DateTimeStamp

Definition of date and time with a resolution of 1 minute and a periodicity of approximately 31 years.

Coded as the number of minute intervals from 01/01/1997, where 01/01/1997 00:00 = 0.

DateTimeStamp ::= INT3

6.19 DayOfWeek

DayOfWeek is a pointer to an entry in a table, held within the CAD and defined in the network, which indicates a day or days of the week during which information, a contract, a product or a ticket is valid or not valid.

DayOfWeek ::= BIT STRING (SIZE(8))

Value Assignment :

The bits are coded each to represent a day. A bit value ‘1’B signifies that the corresponding day is selected.

Multiple bits indicate multiple day selection.

‘abcdefgh’B Selected days :

a Monday,

b Tuesday,

c Wednesday,

d Thursday,

e Friday,

f Saturday,

g Sunday,

h Special day, contract provider specific (such as a public holiday).

6.20 Duration

A measure of a period expressed in units of time. Refer to 6.38 MeasuredParameters for explanation.

Duration ::= Value {time}

6.21 Flag

1 equals true, and zero equals false

Flag ::= BOOLEAN

6.22 HalfDayOfWeek

HalfDayOfWeek is a pointer to an entry in a table, held within the CAD and defined in the network, which indicates a period of a day in the week during which information, a contract, a product or a ticket is valid or not valid.

HalfDayOfWeek ::= BIT STRING (SIZE (16))

The bits are coded each to represent a day. A bit value ‘1’B signifies that the corresponding day is selected.

Multiple bits indicate multiple day selection.

‘abcdefghijklmnop’B Selected days:

- a Monday first period,
- b Monday second period,
- c Tuesday first period,
- d Tuesday second period,
- e Wednesday first period,
- f Wednesday second period,
- g Thursday first period,
- h Thursday second period,
- i Friday first period,
- j Friday second period,
- k Saturday first period,
- l Saturday second period,
- m Sunday first period,
- n Sunday second period,
- o Special day first period,
- p Special day second period,

first and second periods are network specific, special days are network specific.

6.23 HalfDayType

HalfDayType is a pointer to an entry in a table, held within the CAD and defined in the network, which indicates a period of a day of the week during which information, a contract, a product or a ticket is valid or not valid.

HalfDayType ::= BIT STRING (SIZE (7))

The bits are coded each to represent a day-type. A bit value i’B signifies that the corresponding day-type is selected.

Multiple bits indicate multiple day-type selection.

‘abcdefg’B Selected days:

a Saturday first period,

b Sunday first period,

c Weekdays first period,

d Saturday second period,

e Sunday second period,

f Weekdays second period,

g Public holidays

first and second periods are network specific, public holidays are network specific.

6.24 IAI

The Individual Account Identification according to ISO/IEC 7812-1:2006 and including the check digit but held as a single integer.

IAI ::= OCTET STRING(SIZE(12))

Value Assignment : According to ISO/IEC 7812-1

6.25 IIN

The Identification of a company shall be according to ISO/IEC 7812-1.

IIN ::= BCDString (SIZE(6))

6.26 InstancePointer

Pointer to where a data structure may be found. Whilst usage of this element is mandatory, a zero-value indicates that the data structure is void.

InstancePointer{INTEGER:instancePointerRange} ::= INTEGER(0..instancePointerRange)

6.27 INT1

Numerical value 1 byte long

INT1 ::= INTEGER (0..255)

6.28 INT2

Numerical value 2 bytes long

INT2 ::= INTEGER (0..65535)

6.29 INT3

Numerical value 3 bytes long

INT3 ::= INTEGER (0..16777215)

6.30 INT4

Numerical value 4 bytes long

INT4 ::= INTEGER (0..4294967295)

6.31 INTM

Numerical value 6 bits long

INTM ::= INTEGER (0..63)

6.32 INTP

Numerical value 5 bits long

INTP ::= INTEGER (0..31)

6.33 INTS

Numerical value 4 bits long

INTS ::= INTEGER (0..15)

6.34 LanguageAlpha

The alphabetic reference to a language shall be in accordance with the conventional identification of languages as defined in ISO 639-2/B bibliographic code (ASCII).

LanguageAlpha ::= PrintableString (SIZE(3))

NOTE LanguageAlpha is not equivalent to a similar ISO/IEC 7816-6 data element.

6.35 LanguageId

Identifies a language which shall be used in communications with the card holder, where it is practical.

```
LanguageId ::= CHOICE {
    languageAlpha [0] LanguageAlpha,
    languageCode [1] LanguageCode
}
```

6.36 Length

Linear extent of an object or measure of distance expressed in a unit of length. Refer to 6.38 MeasuredParameters for explanation.

Length ::= Value {length}

6.37 MappingType

Card model and mapping (structure) used. It identifies the real mapping.

MappingType ::= INTS

Value assignment : Application specific

6.38 MeasuredParameters

The aim of this concept is to use the same data types for parameters measured in different units. Both a value of a parameter measured in a specific unit and an interval between two values are specified. The ASN.1 listing is as follows:

Value {MeasuredParameterType:param} ::= SEQUENCE {

unit MEASUREDPARAMETER{param}.&Unit

({DefinedMeasuredParameters}),

value INTEGER

}

Interval {MeasuredParameterType:param} ::= SEQUENCE {

lowerBound Value {param} OPTIONAL,

upperBound Value {param} OPTIONAL

}

-- If the "lowerBound" is not present the class starts from 0.

-- If the "upperBound" is not present the class can be increase indefinitely.

MEASUREDPARAMETER {MeasuredParameterType:param} ::= CLASS {

&measuredParameterType MeasuredParameterType(param), &MeasuredParameterUnit

}

WITH SYNTAX {&measuredParameterType &MeasuredParameterUnit}

MeasuredParameterType ::= ENUMERATED {

number (0),

time (1),

length (2),

speed (3),

weight (4),

```
payment       (5),
capacity      (6) }
DefinedMeasuredParameters MEASUREDPARAMETER ::= {
{number NumberUnit} |
{time TimeUnit} |
{length LengthUnit} |
{speed SpeedUnit} |
{weight WeightUnit} |
{payment PaymentUnit} |
{capacity CapacityUnit}
}
```

6.39 Name

Data type to denote a person, a location, an equipment, etc.

```
Name ::= UTF8String (SIZE(0..39))
```

This definition is not identical to the definition of name in ISO/IEC 7816-6:2004, providing the added benefit of UTF8-coding.

6.40 NetworkAccess

Unique identifier for a communication device (e.g. telephone number, fax number).

```
NetworkAccess ::= SEQUENCE {
    protocol            COMMUNICATIONMEDIUM.&protocol ({DefinedMedia}),
    service             COMMUNICATIONMEDIUM.&service ({DefinedMedia} {@.protocol}),
    networkAddress      COMMUNICATIONMEDIUM.&AddressType
                ({DefinedMedia} {@.protocol}) OPTIONAL,
    linkedProtocol      NetworkAccess OPTIONAL
    }
COMMUNICATIONMEDIUM ::= CLASS {
    &protocol      CommunicationProtocol,
    &service CommunicationService,
    &AddressType
```

```
    }
    WITH SYNTAX {&communicationProtocol &communicationService &AddressType}
CommunicationProtocol ::= ENUMERATED {
    gsm             (0),
    ip              (1),
    wap             (2),
    terrestrial     (3),
    ...
    }
CommunicationService ::= ENUMERATED {
    sms  (0),
    gprs      (1),
    tcp       (2),
    udp       (3),
    ...
    }
DefinedMedia COMMUNICATIONMEDIUM ::= {
    {gsm sms PhoneNumber} |
    {gsm gprs PhoneNumber} |
    {internetProtocol tcp IpAddress} |
    {internetProtocol udp IpAddress} |
    {wap NULL IpAddress} |
    {terrestrial NULL PhoneNumber} , ...
    }
IpAddress ::= OCTET STRING
PhoneNumber ::= SEQUENCE OF NumericString       --Including the country code
```

6.41 NetworkId

Identification of a transportation network.

```
NetworkId ::= BIT STRING(SIZE(24))
```

Value Assignment : IIN or OID or ‘cccc cccc cccc rrrr rrrr rrrr’B

‘cccc cccc cccc’B : the country code BCD coded, according to EN ISO 3166-1 Numeric code ‘rrrr rrrr rrrr’B : the network in the country

6.42 NetworkSpecificCompanyId

A constructed data type that uniquely identifies a company or other organisation in the context of its country and attributed network.

NetworkSpecificCompanyId ::= SEQUENCE {

network NetworkId,

company CompanyId

}

Value Assignment : NetworkId specific value.

6.43 Number

Number is a value in 10 power exponent of number unit. Refer to 6.38 MeasuredParameters for explanation.

Number ::= Value {number}

6.44 NumberUnit

The 10-power exponent for the connected value. If X=value Y=exponent, correct interpretation of number is $Xx10^{Y}$.

NumberUnit ::= INTEGER(0..255) -- *Exponent of the power of 10*

6.45 ObjectIdentifier

Identifier of an object defined according to ISO/IEC 8824-1 (OID)

For EN 1545: 1 3 0001 0 01545

6.46 Payment

Payment is a value expressed in a unit of payment. Refer to 6.38 MeasuredParameters for explanation.
Payment ::= Value {payment}

6.47 PayUnitMap

A space saving method of storing currency code within a contract or product. The currency codes defined are subservient to either IIN or NetworkID. It is anticipated that a card accepting device will be equipped to interpret currency code according to IIN or NetworkID.

PayUnitMap ::= BitMap (SIZE(4))

Bit	3	2	1	0
Code Bit	Bit 3	Bit 2	Bit 1	Bit 0

Definition of Currency code, bits 0 and 1:

#	Bit 1	Bit 0	Currency definition
0	0	0	Currency1
1	0	1	Currency2
2	1	0	Currency3
3	1	1	Currency4

Definition of Scaling factor, bits 2 and 3:

#	Bit 3	Bit 2	Scaling factor
0	0	0	X1
1	0	1	X10
2	1	0	X100
3	1	1	X1000

The scaling factor shall be multiplied by the value register to which the currency code definition applies for the purposes of determining the true value represented by the value register.

6.48 Percentage-0

Percentage by step of 1 per cent.

Percentage-0 ::= INTEGER(0..100)

6.49 Percentage-1

Percentage by step of 0,1 per cent.

Percentage-1 ::= INTEGER(0..1000)

6.50 Percentage-2

Percentage by step of 0,01 per cent.

Percentage-2 ::= INTEGER(0..10000)

6.51 PeriodOfDay

Period Of Day is a pointer to an entry in a table, held within the CAD and defined in the network, which identifies the period start and end times of non-validity during which information, a contract, a product or a ticket is valid or not valid.

PeriodOfDay ::= INT1

6.52 Permission

Indicates a permission regarding use of contract. This inverts the function of restriction.

Permission ::= INT1

6.53 PointerValue

Variable length field, comprises the value identified by the PointerQualifierCode.

PointerValue ::= OCTET STRING (SIZE(1..16))

6.54 PTag

An ASN.1 tag that identifies the type of data in an associated data element, data group or data structure. It identifies the specific type such as a contract or a purse.

PTag {INTEGER : pTagRange}::= INTEGER (0.. pTagRange)

6.55 Quantity

The number of items.

Quantity {INTEGER: quantityRange } ::= INTEGER(0..quantityRange)

6.56 ReferenceIdentifier

A string to identify a unique object (e.g. contract, receipt, event, vehicle, stop, line...).

The string is unique within a specified system of reference.

ReferenceIdentifier ::= OCTET STRING (SIZE (referenceIdentifierRange))

6.57 ReferenceNumber

A number to identify a unique object (e.g. contract, receipt, event, ...).

The number is unique within a specified system of reference.

ReferenceNumber {INTEGER : referenceNumberRange}::= INTEGER (0..referenceNumberRange)

Value Assignment : Network Specific

6.58 Restriction

Indicates a restriction on use of a contract.

Restriction ::= INT2

6.59 SequenceNumber

The sequence number is a number that will increment by 1 (one) each time that the number is updated or written to the card. Properly used this number will make it possible to detect missing transactions and contracts in the back-office system.

SequenceNumber {INTEGER : sequenceNumberRange}::= INTEGER (0..sequenceNumberRange)

Value Assignment : Network Specific

6.60 ShortName

Data type to denote a person, a location, an equipment, etc... in a short format not automatically derived from name to save space on e.g. printouts.

ShortName ::= UTF8String (SIZE(8))

6.61 SignedAmount

The value of a monetary transaction. Note that a signedamount can be positive, negative or zero.

SignedAmount always includes VAT where applicable.

SignedAmount {INTEGER: signedMinValueAmountRange, INTEGER : signedMaxValueAmountRange} ::=

INTEGER (signedMinValueAmountRange.. signedMaxValueAmountRange)

Value Assignment **: Unspecified**

6.62 SignedInteger1

SignedInteger1 ::= INTEGER (-127.. 127)

Value Assignment **: Unspecified**

6.63 SignedInteger2

SignedInteger2 ::= INTEGER (-32767.. 32767)

Value Assignment **: Unspecified**

6.64 SignedInteger3

SignedInteger3::= INTEGER (-83886077.. 83886077)

Value Assignment : Unspecified

6.65 Speed

Speed is a value expressed in a unit of speed. Refer to 6.38 MeasuredParameters for explanation.

Speed ::= Value {speed}

6.66 TimeCompact

Formatted time field, where the time is expressed in hours, minutes and seconds.

TimeCompact ::= BIT STRING (SIZE(16))

Value Assignment - Octet Aligned : ‘hhhhhmmmmmmsssss’B (16 bits)

‘hhhhh’B Number of hours

‘mmmmmm’B Number of minutes

‘sssss’B Number of every 2nd second.

‘0000000000000000’B denotes explicitely not time.

6.67 TimeMeasure

TimeMeasure is a value expressed in units of time. Refer to 6.38 Measured Parameters for explanation.

TimeMeasure ::= Value {time}

6.68 TimeReal

Code for a combined date and time field, where the date and time are expressed as seconds past midnight on 1 January 1997 UST.

TimeReal {INTEGER : timeRealRange} ::= INTEGER(0..timeRealRange)

Value Assignment - Bit aligned

Number of seconds since midnight 1 January 1997 ust (30 bits).

6.69 TimeStamp

Number of minutes after midnight, where midnight is time 0.

TimeStamp ::= BIT STRING (SIZE(11))

Value Assignment : ‘mmmmmmmmmmm’B (11 bits)

6.70 VehicleNumber

The official identification of a vehicle as defined in ISO 14816.

VehicleNumber ::= LicPlateNumber

6.71 VersionNumber

Identifies the version of a data structure.

VersionNumber {INTEGER: versionNumberSize}

::= BIT STRING (SIZE (versionNumberSize))

Value Assignment : Octet aligned, either numbered 0..255 or using 4 bits to indicate 'release' and 4 bits to indicate 'revision' ('10'H reflects the initial version 1.0)

6.72 Weight

Weight of an object or measure of weight. Weight is a value expressed in a unit of weight. Refer to 6.38 MeasuredParameters for explanation.

Weight ::= Value {weight}

7 Data elements with associated code lists

7.1 General

These are data elements expressed as universal ASN.1 types, with an associated mandatory code list.

7.2 CapacityUnit

The unit in which a value of volume, or occupancy of either a vehicle or a cargo consignment is expressed.

```
CapacityUnit ::= INT1
    --iso7372PackageCode7065        (0..153),
    --rfuCEN                        (154..175),
    --volumeNotFurtherSpecified     (176),
    --volumeDecilitres              (177),
    --volumeLitres                  (178),
    --volumeCubicMetres             (179),
    --volumeCountrySpecificUse      (180),
    --networkIdSpecific             (181..191),
    --livestocknotFurtherSpecified  (192),
    --rfuCEN                        (193..200),
    --livestockCountrySpecificUse   (201),
    --networkIdSpecific             (202..207),
    --countrySpecific               (208..255)
```

7.3 CommercialTransportProductCode

A code identifying the product sold to the customer in the environment of commercial transport.

```
CommercialProductCode ::= INT1
```

--System-fee	(1)	Shipment-cost	(2)
--Diesel	(3)	Diesel-bio	(4)
--Red-diesel	(5)	Petrol	(6)
--Petrol-unleaded	(7)	Super	(8)
--Super-unleaded	(9)	Super-plus	(10)
--Two-stroke-mixture	(11)	Fuel-98	(12)
--Super-unleaded-power	(13)	Super-plus-racing	(14)
--Liquified-petroleum-gas	(15)	Motor-oil	(16)
--Motor-oil-II	(17)	Transmission-oil	(18)
--Batteries	(19)	Maps	(20)
--Other-accessories	(21)	Other-accessories–red-VAT	(22)
--Other-accessories-red-VAT-II	(23)	Other-accessories-VAT-free	(24)
--Car-wash	(25)	Vehicle-services	(26)
--Testing-services	(27)	Miscellaneous-station	(28)
--Miscellaneous-station-red-VAT	(29)	Miscellaneous-station-VAT-free	(30)
--Tyres	(31)	Tyres VAT free	(32)
--Tyres-delivered	(33)	Tyres-delivered-VAT-free	(34)
--Tyre-fitting	(35)	Tyre-fitting-VAT-free	(36)

--Tyre-fitting-mobile	(37)	Tyre-fitting-mobile-VAT-free	(38)
--Parts-tyres	(39)	Parts-tyres-VAT-free	(40)
--Repair	(41)	Repair VAT free	(42)
--Miscellaneous-garage	(43)	Miscellaneous-garage-red-VAT	(44)
--Miscellaneous-garage-VAT-free	(45)	Parts	(46)
--Parts-red-VAT	(47)	Parts-VAT-free	(48)
--Parts-sales	(49)	Parts-sales-VAT-free	(50)
--Parts-sales-red-VAT	(51)	Diesel-to-garage	(52)
--Towing	(53)	Towing-VAT-free	(54)
--Insurance	(55)	Insurance-VAT-free	(56)
--Cooling-system-service	(57)	Other-service	(58)
--Other-service-red-VAT	(59)	Other-service-VAT-free	(60)
--Labour	(61)	Labour-red-VAT	(62)
--Labour-VAT-free	(63)	Toll	(64)
--Toll-red-VAT	(65)	Toll-VAT-free	(66)
--Tunnel-fee	(67)	Tunnel-fee-red-VAT	(68)
--Tunnel-fee-VAT-free	(69)	Bridge-toll	(70)
--Bridge-toll-red-VAT	(71)	Bridge-toll-VAT-free	(72)
--Entry-fee	(73)	Service-for-road-tax	(74)
--HGV-tax	(75)	Combined-traffic	(76)
--Combined-traffic-VAT-free	(77)	Euro-vignette	(78)
--Vignette-superior-to-3,5-t	(79)	Vignette-AUT-inferior-or-equal-to-12-t	(80)
--Transport-accompaniment	(81)	Cleaning	(82)
--Cleaning-VAT-free	(83)	Mobile-cardware	(84)
--Mobile-hardware	(85)	Rental-service	(86)
--Rental-service-red-VAT	(87)	Parking-area-charge	(88)
--Ferry-charge	(89)	Ferry-charge-VAT-free	(90)
--Ferry-charges-inland-waters	(91)	Boarder-clearance	(92)
--Boarder-clearance-VAT-free	(93)	Loan-payment-VAT-free	(94)
--Repair-breakdown	(95)	Repair-breakdown-VAT-free	(96)
--Towing-breakdown	(97)	Towing-breakdown-VAT-free	(98)
--Delivery-costs	(99)	Delivery-costs-VAT-free	(100)
--Transport-insurance	(101)	Transport-insurance VAT-free	(102)
--Tyres-breakdown	(103)	Tyres-breakdown-VAT-free	(104)
--Replacement-vehicle	(105)	Replacement-vehicle-VAT-free	(106)
--Intervention	(107)	Intervention-red-VAT	(108)
--Intervention-VAT-free	(109)		
--RfuCEN	(110.		

	.180)
--NetworkIdSpecific	(181. .255)

NOTE “red” means reduced.

7.4 ConditionCode

A code specifying the condition for the exchange of a travel contract for an identical contract on a different travel service or a refund.

```
ConditionCode ::= ENUMERATED {
```

not-exchangeable	(0),
before-departure	(1),
up-to-1-hour-before-departure	(2),
up-to-2-hours-before-departure	(3)
up-to-1-day-before-departure	(4)
up-to-2-days-before-departure	(5)
up-to-1-week-before-departure	(6)
up-to-2-weeks-before-departure	(7)
up-to-1-month-before-departure	(8)
up-to-2-months-before-departure	(9)
up-to-1-hour-after-departure	(10)
up-to-2-hours-after-departure	(11)
up-to-1-day-after-departure	(12)
up-to-2-days-after-departure	(13)
up-to-1-week-after-departure	(14)
up-to-2-weeks-after-departure	(15)
up-to-1-month-after-departure	(16)
up-to-2-months-after-departure	(17)
rfuCEN1	(18)
rfuCEN2	(19)
rfuCEN3	(20)
rfuCEN4	(21)
rfuCEN5	(22)
rfuCEN6	(23)
rfuCEN7	(24)
networkIdSpecific1	(25)
networkIdSpecific2	(26)

networkIdSpecific3 (27)
networkIdSpecific4 (28)
networkIdSpecific5 (29)
networkIdSpecific6 (30)
networkIdSpecific7 (31)
}

7.5 DayOfValidityCode

A code indicating the validity of a contract upon the day of issue, and upon the day of expiry of that contact.

```
DayOfValidityCode::= INTS
--contract-is-not-valid-on-the-day-of-expiry    (0),
--contract-is-valid-on-the-day-of-expiry        (1),
--contract-is-not-valid-on-the-day-of-issue     (2),
--contract-is-valid-on-the-day-of-issue         (3),
--rfuCEN                                        (4..15)
```

7.6 DestinationOrOriginCode

It indicates whether a location identity data element contains a destination or an origin.

```
DestinationOrOriginCode ::= ENUMERATED {
    origin      (0),
    destination (1)
}
```

7.7 DeviceTypeCode

A code indicating the type of a CAD.

```
DeviceTypeCode ::= INTS
      --otherwise-specified                 (0)
      --registrationTerminal                (1), (used for cico/bibo etc.)
      --ticketVendingMachine                (2),
      --inspectionTerminal                  (3),
      --customerMediaIssuingTerminal        (4),
      --customerMediaReturnTerminal         (5),
      --validator                           (6),
      --multiFunctionTerminal               (7),
      --informationTerminal                 (8),
      --loadingTerminal                     (9),
      --rfuCEN                              (10..12),
      --networkIdSpecific                   (13..15)
```

7.8 DirectionCode

A code indicating the direction of travel of the vehicle or train.

```
DirectionCode ::= ENUMERATED {
    outward    (0),
    inward     (1)
}
```

7.9 EntitlementTypeCode

A code identifying the nature of an entitlement. These are attributes of the relationship between the holder and the contract that lead to discount percentages and/or benefits.

```
EntitlementTypeCode ::= ENUMERATED {
        no-entitlement          (0),
        warrant-limited         (1),
        free-ride               (2),
        proportional-fare       (3),
        flat-fare-discount      (4),
        flat-fare               (5),
        charge-to-account       (6),
        subscription            (7),
        frequent-traveller      (8),
        senator                 (9),
        premium                (10),
        gold-status            (11),
        silver-status          (12),
        capped-fare            (13),
        free-travel            (14),
        half-fare              (15),
        rfuCEN1                (16),
        rfuCEN2                (17),
        rfuCEN3                (18),
        rfuCEN4                (19),
        rfuCEN5                (20),
        rfuCEN6                (21),
        rfuCEN7                (22),
        rfuCEN8                (23),
        rfuCEN9                (24),
        networkIdSpecific1     (25),
        networkIdSpecific2     (26),
        networkIdSpecific3     (27),
```

```
        networkIdSpecific4      (28),
        networkIdSpecific5      (29),
        networkIdSpecific6      (30),
        networkIdSpecific7      (31)
}
```

7.10 EventTypeCode

A code identifying the nature of an event.

```
EventTypeCode ::= ENUMERATED {
not-specified                                   (0),
sale                                            (1),
validation-outward-journey-if-return-ticket     (2),
undo-previous-event-without-refund              (3),
str-load                                        (4),
str-autoload                                    (5),
validation-return-journey                       (6),
str-debit                                       (7),
exchange                                        (8),
redeem-loyalty-points                           (9),
undo-previous-event-with-refund                 (10),
check-in                                        (11),
check-out                                       (12),
activate-stored-ticket                          (13),
record-of-multiple-leg-journey                  (14),
cta-payment-received                            (15),
check-in-transfer                               (16),
be-in-transfer                                  (17),
user-modification                               (18),
consumed                                        (19),
marked-as-blocked                               (20),
undo-blocking                                   (21),
be-in                                           (22),
be-out                                          (23),
interruption                                    (24),
refund-authorised                               (25),
inspection                                      (26),
in-out                                          (27),
undo-validation                                 (28),
networkIdSpecific1                              (29),
```

networkIdSpecific2 (30),

networkIDSpecific3 (31),

Explanations of the codes:

- validation means that the contract is marked as in use by time stamping. Validity checks may be made as a part of this process;

- consumed means that the contract is marked as in use or has been used and shall not be used again;

- undo-validation means to reverse the validation process to re-instate the contract (e.g. if after validation the service cannot be provided);

- interruption indicates that the service was only partially provided;

- exchange means change of service elements (such as reservations) without changing the terms of the underlaying contract.

7.11 GenderCode

A code identifying the sexual identity of the holder according to ISO/IEC 5218. GenderCode ::= BIT STRING (SIZE(2))

-- '00'B: not known

-- '01'B: male

-- '10'B: female

-- '11'B: unspecified

7.12 HotListStatusCode

A code used in hotlists, whitelists and statuslists that indicates the general validity of a device/card/application/product to the rest of the system.

```
HotListStatusCode ::= ENUMERATED {
    ok                      (0)
    blocked-Undefined       (1)
    blocked-Stolen          (2)
    blocked-Lost            (3)
    blocked-Refunded        (4)
    blocked-OverLimit       (5)
    monitored               (6)
    auto-reload-disabled    (7)
    unblocked               (8)
    blocked-misuse          (9)
    rfuCEN1                 (10)
    rfuCEN2                 (11)
    rfuCEN3                 (12)
    networkIdSpecific1      (13)
```

networkIdSpecific2 (14)

networkIdSpecific3 (15)

}

7.13 LanguageCode

Numeric reference to a language, in accordance with the conventional identification of languages as defined in ISO 639-2/B bibliographic code (ASCII).

```
LanguageCode ::= INT1
-- Undefined                                (0),
-- Abkhazian; Abkhaz;                       ab (1),
-- Afan Oromo; Oromo; Galla                 om (2),
-- Afar                                     aa (3),
-- Afrikaans                                af (4),
-- Akan                                     ak (5),
-- Albanian                                 sq (6),
-- Amharic                                  am (7),
-- Arabic                                   ar (8),
-- Armenian                                 hy (9),
-- Assamese                                 as (10),
-- Avar; Avarish                            av (11),
-- Avestan                                  ae (12),
-- Aymara                                   ay (13),
-- Azerbaijani                              az (14),
-- Bambara                                  bm (15),
-- Bashkir                                  ba (16),
-- Basque                                   eu (17),
-- Belarusian                               be (18),
-- Bengali; Bangla                          bn (19),
-- Bhutani; Butanese; Dzongkha              dz (20),
-- Bihari                                   bh (21),
-- Bislama                                  bi (22),
-- Bosnian                                  bs (23),
-- Breton                                   br (24),
-- Bulgarian                                bg (25),
-- Burmese; Myanmar                         my (26),
-- Cambodian; Khmer                         km (27),
-- Castilian; Spanish                       es (28),
-- Catalan                                  ca (29),
-- Chamorro                                 ch (30),
```

-- Chechen	ce (31),
-- Chichewa; Chewa; Nyanja	ny (32),
-- Chinese	zh (33),
-- Chuang; Zhuang	za (34),
-- Church Slavonic; Church Slavic; Old Slavonic;	
-- Old Church Slavonic; Old Bulgarian	cu (35),
-- Chuvash	cv (36),
-- Cornish	kw (37),
-- Corsican	co (38),
-- Cree	cr (39),
-- Croatian	hr (40),
-- Czech	cs (41),
-- Danish	da (42),
-- Dutch	ni (43),
-- English	en (44),
-- Esperanto	eo (45),
-- Estonian	et (46),
-- Ewe	ee (47),
-- Faroese; Faeroese	fo (48),
-- Fijian	fj (49),
-- Finnish	fi (50),
-- French	fr (51),
-- Frisian	fy (52),
-- Fulah; Fula; Fulani; Fulfulde; Peul	ff (53),
-- Gaelic; Scottish Gaelic	gd (54),
-- Galician; Gallegan	gi (55),
-- Ganda; Luganda	lg (56),
-- Georgian	ka (57),
-- German	de (58),
-- Gikuyu; Kikuyu	ki (59),
-- Greenlandic; Kalaallisut	kl (60),
-- Guarani	gn (61),
-- Gujarati	gu (62),
-- Hausa	ha (63),
-- Hebrew	he (64),
-- Herero	hz (65),
-- Hindi	hi (66),
-- Hiri Motu	ho (67),
-- Hungarian	hu (68),

-- Icelandic	is (69),
-- Ido	io (70),
-- Igbo	ig (71),
-- Indonesian	id (72),
-- Interlingue	ie (73),
-- Irish	ga (74),
-- Italian	it (75),
-- Japanese	ja (76),
-- Javanese	jv (77),
-- Kannada	kn (78),
-- Kanuri	kr (79),
-- Kashmiri	ks (80),
-- Kazakh	kk (81),
-- Kikuyu;Gikuyu	ki (82),
-- Kinyarwanda; Rwanda	rw (83),
-- Kirundi; Rundi	rn (84),
-- Kiswahili; Swahili	sw (85),
-- Komi	kv (86),
-- Kongo	kg (87),
-- Korean	ko (88),
-- Kurdish	ku (89),
-- Kwanyama; Kuanyama	kj (90),
-- Kyrgyz; Kirghiz	ky (91),
-- Laotian; Lao	lo (92),
-- Latin	la (93),
-- Latvian	lv (94),
-- Lingala	ln (95),
-- Lithuanian	lt (96),
-- Interlingua	ia (97),
-- Inuktitut	iu (98),
-- Inupiaq	ik (99),
-- Luba-Katanga	lu (100),
-- Luganda; Ganda	lg (101),
-- Luxembourgish	lb (102),
-- Macedonian	mk (103),
-- Malagasy	mg (104),
-- Malay	ms (105),
-- Malayalam	ml (106),
-- Maldivian; Divehi	dv (107),

-- Maltese	mt (108),
-- Manx	gv (109),
-- Maori	mi (110),
-- Marathi	mr (111),
-- Marshallese	mh (112),
-- Modem Greek (post 1453)	el (113),
-- Moldavian	mo (114),
-- Mongolian	mn (115),
-- Nauruan	na (116),
-- Navajo; Navaho	nv (117),
-- Ndonga	ng (118),
-- Nepali	ne (119),
-- North Ndebele	nd (120),
-- Northern Sami	se (121),
-- Norwegian	no (122),
-- Norwegian Bokmál	nb (123),
-- Norwegian Nynorsk	nn (124),
-- Occitan; Provençal (post 1500)	oc (125),
-- Ojibwa	oj (126),
-- Oriya	or (127),
-- Ossetian; Ossetic	os (128),
-- Pali	pi (129),
-- Pashto; Pushto	ps (130),
-- Persian; Farsi	fa (131),
-- Polish	pl (132),
-- Portuguese	pt (133),
-- Punjabi; Panjabi -	pa (134),
-- Quechua	qu (135),
-- Rhaeto-Romance	rm (136),
-- Romanian	ro (137),
-- Russian	ru (138),
-- Rwanda; Kinyarwanda	rw (139),
-- Samoan	sm (140),
-- Sango; Sangho	sg (141),
-- Sanskrit	sa (142),
-- Sardinian	sc (143),
-- Serbian	sr (144),
-- Serbo-Croatian	sh (145),
-- Sesotho; Southern Sotho	st (146),

-- Setswana; Tswana	tn (147),
-- Shona	sn (148),
-- Sindhi	sd (149),
-- Sinhala; Sinhalese; Singhalese	si (150),
-- Slovak	sk (151),
-- Slovenian	sl (152),
-- Somali	so (153),
-- South Ndebele	nr (154),
-- Spanish; Castilian	es (155),
-- Sundanese	su (156),
-- Swahili; Kiswahili	sw (157),
-- Swazi; Swati; Siswati	ss (158),
-- Swedish	sv (159),
-- Tagalog	tl (160),
-- Tahitian	ty (161),
-- Tajiki	tg (162),
-- Tamil	ta (163),
-- Tatar	tt (164),
-- Telugu	te (165),
-- Thai	th (166),
-- Tibetan	bo (167),
-- Tigrinya	ti (168),
-- Tongan (Tonga Islands)	to (169),
-- Tsonga	ts (170),
-- Turkish	tr (171),
-- Turkmen	tk (172),
-- Twi	tw (173),
-- Uighur	ug (174),
-- Ukrainian	uk (175),
-- Urdu	ur (176),
-- Uzbek	uz (177),
-- Venda	ve (178),
-- Vietnamese	vi (179),
-- Volapuk	vo (180),
-- Waltoon	wa (181),
-- Welsh	cy (182),
-- Wolof	wo (183),
-- Xhosa	xh (184),
--- Yiddish	yi (185),

-- Yoruba yo (186),
-- Zulu zu (187),
-- rfuCEN (188..255)

7.14 LegislationCode

A reference to the legislative convention governing the services (e.g. CIV).

```
LegislationCode ::= ENUMERATED {
    none                (0),
    civ                 (1),
    rfuCEN1             (2),
    rfuCEN2             (3),
    rfuCEN3             (4),
    networkIdSpecific   (5..7)
}
```

7.15 LengthUnit

Unit in which a length (as linear extent or distance) is expressed. LengthUnit ::= ENUMERATED {

```
  milliMetre      (0)
  centiMetre      (1)
  deciMetre       (2)
  metre           (3)
  tenMetres       (4)
  hundredMetres   (5)
  kiloMetre       (6)
  miles           (7)
}
```

7.16 LocationQualifierCode

A code identifying the means by which a location is specified.

```
LocationQualifierCode ::= INT1
    --alphanumericAndSpecialCharacters      (0),
    --GlobalPositioningSystem               (16),
    --geographicalDataFile                  (32),
    --RadioDataSystem                       (48),
    --GlobalSystemforMobileCommunication    (64),
    --lowOrbiter                            (80),
    --DedicatedShortRangeCommunication      (96),
    --GNSS                                  (112),
    --Galileo                               (128),
    --UIC                                   (144),
    --SNCF                                  (145),
```

```
    --IATA                          (146),
    --mutuallyAgreed                (147  159),
    --rfuCEN                        (160  199),
    --networkIdSpecific             (200   255)
```

7.17 LocationTypeCode

A code specifying the business type of a location.

```
LocationTypeCode ::= INT1
    -- busStopStation               (0)
    -- metroStation                 (1)
    -- railwayStation               (2)
    -- tramStopStation              (3)
    --ferryPort                     (4),
    --tollPlaza                     (5),
    --parkingGarageLotPlace         (6),
    --taxationStation               (7),
    --purseLoadAgent                (8),
    --travelAgent                   (9),
    --bankPostOffice                (10),
    --retailAgent                   (11),
    --tollTunnel                    (12),
    --garage                        (13),
    --fuelStation                   (14),
    --customs                       (15),
    --areaID                        (16),
    --corridor                      (17),
    --virtualGantry                 (18),
    --stop-on-request               (19),   between official stops
    --areaBitMapValidInZone         (20),
    --areaBitMapValidToZone         (21),
    --taxiRank                      (22),
    --latitudeLongitude             (23),
    --latitudeLongitudeAltitude     (24),
    --geographicalDataFile          (25),
    --UIC                           (26),
    --IATA                          (27),
    --place-of-work                 (28),
    --school-college-university     (29),
    --mixedTypeStation              (30),
    --rfuCEN                        (31..199),
    --networkIdSpecific             (200..255)
```

7.18 PersonalisationBiometricCode

Identifies the biometric information type used.

```
PersonalisationBiometricCode ::= INTP
    --unspecified                   (0),
    --photograph                    (1),
```

```
    --iris                  (2),
    --fingerprint           (3),
    --dna                   (4),
    --facialGeometry        (5),
    --signature             (6),
    --rfuCEN                (7..15),
    --networkIdSpecific     (16..31)
```

7.19 PersonalisationTypeCode

To identify the type of personalisation of the application (and the presence of the holder structure).

```
PersonalisationTypeCode ::= ENUMERATED {
Anonymous           (0)
Identified          (1), --(information in the back office)
Personalized        (2)
networkIdSpecific   (3…7)
}
```

7.20 PointerQualifierCode

Qualifier indicating the search commencement point and the nature of the search argument.

```
PointerQualifierCode ::= INT1
    --masterFileLevel                       (0),
    --surfaceTransportApplication           (1),
    --selectedDedicatedFilelevel            (2),
    --fullyQualifiedPathSupplied            (3),
    --otherISO/IEC 7816-4-5Environments     (4.. 15),
    --dedicatedFileNameSupplied             (16),
    --twoOctetElementaryFileIdSupplied      (17),
    --otherCardSpecificEnvironments         (18..32),
    --absoluteAddress                       (33),
    --relativeOffset                        (34),
    --      otherSequentialStorageEnvironments (35 .. 47),
    --berTagSpecification                   (48),
    --perTagSpecification                   (49),
    --otherASN1Environments                 (50 .. 63),
    --rfuCEN                               (64 .. 255)
```

7.21 PreferenceTypeCode

Identifies the holder related interface requirements according to EN 1332-4.

```
PreferenceTypeCode::= ENUMERATED {
Not-specified                   (0)
use-CEN-ISSS-13987-2003-e-URI   (1)
networkIdSpecific               (2..7)
}
```

7.22 ProfileCodeIOP

Code classifying the customer according to certain criteria. This profile may be used to determine price calculation and/or benefits. Classes may describe the customer (e.g. pensioner) but may also directly refer to the price reduction percentage (e.g. 25 %) applying to the customer.

It is recommended that the Adult(1) & Child(2) codes are not used. The distinction between adult and child should be based upon “date of birth.” {we have left the codes in the list for reasons of backwards compatibility with ENV 1545-1}.

```
ProfileCodeIOP ::= INTM
    --unspecified                   (0),
    --adult                         (1),
    --child                         (2),
    --student                       (3),
    --pensioner                     (4),
    --disabledNotfurtherspecified   (5),
    --disabledVisuallyImpaired      (6),
    --disabledHearingImpaired       (7),
    --unemployed                    (8),
    --staff                         (9),
    --military                      (10),
    --resident                      (11),
    --industrialOwnedHaulage        (12),
    --busTransportCompany           (13),
    --longDistanceTransport         (14),
    --localTransport                (15),
    --commuter                      (16),
    --chargeableAnimal              (17),
    --chargeableObject              (18),
    --scholar                       (19),
    --trainee                       (20),
    --police                        (21),
    --motorbike                     (22),
    --pushbike                      (23),
    --perambulatorWithoutChild      (24),
    --senior                        (25),
    --family                        (26),
    --driver                        (27),
    --rfuCEN                      (28...63)
```

7.23 ProfileCodeNetwork

Code classifying the customer according to certain criteria. This profile may be used to determine price calculation and/or benefits. Classes may describe the customer (e.g. student) but may also directly refer to the price reduction percentage (e.g. 25 %) applying to the customer.

ProfileCodeNetwork contains nationally defined profiles, and would be used only where the terminal recognised the NetworkID and understood the coding.

```
ProfileCodeNetwork ::= INTM
    -- networkIDSpecific      (0..63)
```

7.24 ReferenceTypeCode

A code indicating the type of business object identified by a reference.

```
ReferenceTypeCode ::= ENUMERATED {
unspecified          (0)
contract             (1)
travelDossier        (2)
issueControl         (3)
rfuCEN1              (4)
rfuCEN2              (5)
networkIdSpecific1   (6)
networkIdSpecific2   (7)
}
```

7.25 RestrictTimeCode

This element is retained from ENV 1545, and is retained for the purposes of backwards compatibility. It is replaced by RestrictPeriodOfDay.

```
RestrictTimeCode ::= PeriodOfDay
```

7.26 ResultCode

A code designating whether a transaction was completed successfully.

```
ResultCode ::= BIT STRING (SIZE(8))
```
Value Assignment: Hexadecimal

Most significant 4 bits: 0 OK:

'0x'H OK

Most significant 4 bits > 0 Not OK:

'1x'H Not OK, not specified further

'2x'H Not OK, Abnormal (First or Previous) Event

'3x'H Not OK, Contract not accepted

'4x'H Not OK, Application not accepted

'5x'H Reserved for EFC application 1

'6x'H Reserved for EFC application 2

Least significant 4 bits:

'x0'H not specified further

'x1'H Balance close to zero

'x2'H Balance now negative

'x3'H Balance Overflow

'x4'H Provider not accepted

'x5'H Authentication failure

'x6'H Vehicle Class incorrect

'x7'H Reserved for EFC application 1

'x8'H Reserved for EFC application 2

7.27 RevocationDetailsCode

Code to indicate the reason to refuse a validation or a financial transaction.

```
RevocationDetailsCode ::= ENUMERATED {
revocationReleased (0)
noFundsAvailable (1)
paymentMeansExpired (2)
securityViolation (3)
contractNotConfirmed (4)
hotlisted (5)
rfuCEN (6)
networkIdSpecific (7)
}
```

7.28 RoundingCode

Code to define rounding rules used by the terminal when calculating proportional, discounted or half fares. When no rounding rule is defined then rules implemented in the terminal shall be used.

```
RoundingCode ::= INTS
    --No-rounding-rule-defined                              (0..3),
    --Round-down-to-single-currency-unit                    (4),
    --Round-down-to- nearest-multiple-of-5-currency-units   (5),
     --Round-up-to-single-currency-unit                     (6),
     --Round-up-to- nearest-multiple-of-5-currency-units    (7),
     --rfuCEN                                               (8..12),
     --networkIdSpecific                                    (13..15)
```

7.29 SecurityServicesCode

The SecurityServices Code lists the security services to be used within a session.

```
SecurityServicesCode ::= BIT STRING {
digitalsignature (0)
```

mac (1)
symmetricEncryption (2)
asymmetricEncryption (3)
biometricAuthentication (4)
rfuCEN1 (5)
rfuCEN2 (6)
networkIdSpecific (7)
}

7.30 SeriousnessCode

Value denoting the severity of an event.

```
SeriousnessCode ::= ENUMERATED {
      none          (0),
      information   (1),
      warning       (2),
      fault          (3)
      rfuCEN1       (4),
      rfuCEN2       (5),
      rfuCEN3       (6),
      rfuCEN4        (7)
}
```

7.31 SpeedUnit

Unit which the scalar measure of a rate of movement is expressed as average or instantaneous speed.

```
SpeedUnit ::= ENUMERATED {
    metrespersecond     (0),
    kilometresperhour   (1)
}
```

7.32 StatusCode

Code denoting the status of a contract or application compliant to operating requirements or regulations.

```
StatusCode ::= INTEGER (0..127)
      --never-validated                (0)
      --used-once                      (1)
      --validated                      (2)
      --contract-already-renewed       (3)
      --punched                        (4)
      --cancelled                      (5)
      --interrupted                    (6)
      --ok                             (7)
      --new                            (8)
```

--released	(9)
--corrupted	(10)
--terminated	(11)
--expired	(12)
--not-available-for-validation	(13)
--free-entry (may be re-used)	(14)
--active	(15)
--pre-allocated (as yet unused)	(16)
--completed-and-can-be-removed	(17)
--completed-and-cannot-be-removed	(18)
--blocked	(19)
--data-group-encrypted-flag	(20), (to maintain operator data privacy)
--data-group-anonymous-flag	(21), (to support anonymity requirements)
--rfuCEN	(22..32),
--pending	(33)
--rfuCEN	(34..62),
--suspended	(63),
--rfuCEN	(64..87),
--disabled	(88),
--networkIdSpecific	(89..124),
--suspended-contract	(125),
--invalid-contract	(126),
--invalid-and-reimbursed-contract	(127)

Value Assignment : Numeric values are chosen in order to give «visual» information for some values (according to the ASCII character set) :

"☺" 1 first validation was performed on the contract

"!" 33 Still enabled/suspended but action pending(refer to the related information)

"?" 63 Temporarily disabled/suspended (refer to the related information)

"X" 88 Not OK, disabled/suspended

7.33 TimeUnit

Unit in which the duration between two events or the timing of an event is expressed.

```
TimeUnit ::= ENUMERATED {
```

millisecond	(0)
second	(1)
minute	(2)
hour	(3)
day	(4), --intercalendary day not counted

week (5)

month (6)

year (7)

}

7.34 TransactionModeCode

The transaction mode identifies the moment in time when the price of the service/product is determined. TransactionModeCode ::= BIT STRING (SIZE(2)) tt Transaction mode (2 bits) :

‘00’B : unspecified,

‘01’B : prepriced, (price is determined in advance of travel)

‘10’B : postpriced (price is determined after travel, possibly in back office)

‘11’B : trippriced (price is determined after the end of a trip, i.e. at check out)

7.35 TransportTypeCode

Code indicating the type of transport.

```
TransportTypeCode ::= INTP
```

--not-specified-further (0)

--urbanBus (1)

--interurbanBus (2)

--lightTrainMassTransit (3), (Underground, Metro, LTR)

--tram (4)

--commuterTrain (5)

--waterborneVehicle (6)

--toll (7)

--parking (8)

--taxi (9)

--highSpeedTrain (10)

--ruralBus (11)

--expressCommuterTrain (12)

--paraTransit (13)

--self-drive-vehicle (14)

--coach (15)

--locomotive (16)

--poweredMotorVehicle (17)

--trailer (18)

--regionalTrain (19)

--interCity (20)

--funicularTrain (21)

--cableCar (22)
--selfServiceBicycle (23)
--carSharing (24)
--carPooling (25)
--rfuCEN (26...27),
--networkIdSpecific (29...31)

7.36 UserActionCode

Code identifying a particular action related to card usage.

```
UserActionCode ::= INTP
    -- notSpecifiedFurther      (0),
    -- entry                    (1),
    -- exit                     (2),
    -- passage                  (3),
    -- checkpointInspection     (4),
    -- autonomous               (5),
    -- interchange              (6),
    -- validation               (7),
    -- presenceDetected         (8),
    -- rfuCEN                   (9...19),
    -- networkIdSpecific        (20...31),
```

7.37 WeightUnit

Unit in which the weight is expressed.

```
WeightUnit ::= ENUMERATED {
gramme              (0)
kilogramme          (1)
tenKilogrammes      (2)
hundredKilogrammes  (3)
tonne               (4)
rfuCEN1             (5)
rfuCEN2             (6)
rfuCEN3             (7)
}
```

7.38 UserMediaTypeCode

Code identifying the type of user media used for storing the application

```
UserMediaTypeCode ::= INTM
        -- unspecified                  (0),
```

-- ID-1 card-ISO-IEC 7810	(1),
-- mobilePhone	(2),
--rfuCEN	(3..31)
--networkIDSpecific	(32..63)

7.39 SecurityAlgorithmCode

Code identifying the algorithm used in relation to the SecurityServiceCode used

```
SecurityAlgorithmCode ::= ENUMERATED{
```

not-specified	(0),
RSA	(1),
undefined-asymmetric-algorithm	(2),
rfuCEN1	(3),
AES	(4),
single-DES	(5),
triple-DES-(1key)	(6),
triple-DES-(2key)	(7),
triple-DES-(3key)	(8),
SHA-1	(9),
MD5	(10),
undefined-MAC-algorithm	(11),
rfuCEN2	(12),
undefined-symmetric-algorithm	(13),
rfuCEN3	(14),
elliptic-curve-Diffie-Hellmann	(15),
elliptic-curve-digital-signature-algorithm	(16),
elliptic-curve-integrated-encryption-scheme	(17),
undefined-elliptic-curve-algorithm	(18),
rfuCEN4	(19),
networkIDSpecific	(20..31)

}

8 General data elements

8.1 AccountingId

Unique reference set by the entity that loads the travel contract on the smart card, to be used in settlement documentation between this entity and the service provider.

AccountingId ::= ReferenceIdentifier

Value Assignment : a NetworkId specific value

8.2 ActionListSequenceNumber

A number used to prevent multiple actioning of action list items.

ActionListSequenceNumber ::= SequenceNumber

Value Assignment : a NetworkId specific value.

8.3 AlgorithmId

Identifies the algorithm used.

AlgorithmId ::= INT1

Value Assignment : a NetworkId specific value.

8.4 ApplicationId

The reference for an application shall be as defined in ISO/IEC 7816-5 (AID).

ApplicationId ::= AID

8.5 ApplicationOwner

Identifies the actor that is responsible for the transport application on this medium.

ApplicationOwner ::= NetworkSpecificCompanyId

8.6 BirthDate

BirthDate is the date of birth of a person.

BirthDate ::= Datef

8.7 BirthName

The name given to a person at birth.

BirthName ::= Name

8.8 BirthPlace

The location of birth of the person

BirthPlace ::= Name

8.9 CollectionAndForwardingOperator

Actor that collects the usage and revenue data, receives data from other collection operators and forwards the “not on us data” to other collection operators.

CollectionAndForwardingOperator ::= NetworkSpecificCompanyId

8.10 CompanyName

The name of the company.

CompanyName ::= Name

8.11 ContractDependencyPointer

The pointer to another contract or product which provides other useful informations.

ContractDependencyPointer ::= InstancePointer

8.12 ContractTypesAllowed

Identifies the contract types which may be loaded to an application by reference to a networkId specific table stored in the CAD. Each line in this table will identify a contract type or a group of contract types. Value of zero indicates that all contracts are allowed.

ContractTypesAllowed ::= ReferenceNumber

8.13 CustomerContractProvider

CustomerContractProvider is the actor with the obligation to provide the service specified in the contract. He also holds the centrally held account.

CustomerContractProvider ::= NetworkSpecificCompanyId

8.14 CustomerNumber

The referencenumber of a customer.

CustomerNumber ::= ReferenceNumber

8.15 Date

Data element to specify a general date.

Date ::= DateCompact

8.16 DateTime

Data element to specify a general time.

DateTime ::= DateTimeCompact

8.17 DateTimeBand

Defines product specific validity date and times by reference to a contract provider specific table held in the CAD.

DateTimeBand ::= ReferenceNumber

8.18 DeductionPercentage

The percentage to represent a fare reduction.

DeductionPercentage ::= Percentage-0

8.19 DelayCounter

A means of recording the delay to a service in the card defined in minutes.

DelayCounter ::= Counter

8.20 DeviceId

Identifier for a card accepting device. The device identifier will be used in back-office procedures and need only be significant to the device-managing company. The company's partners in an integrated scheme need to read this identifier but not necessarily to interpret it.

DeviceId ::= ReferenceIdentifier

8.21 DisplayMessageNumber

Reference number of a message which shall be displayed.

DisplayMessageNumber ::= ReferenceNumber

8.22 EmailAddress

An e-mail address.

EmailAddress ::= NetworkAccess

8.23 EndDate

EndDate is the final date in a period of days, e.g. it is the last day when a contract is valid. Period ends at 24 h on the end date unless an end time is explicitly specified (acceptance after this time is at the discretion of the service provider). May be used for ExpiryDate etc.

EndDate ::= DateCompact

8.24 EndDatePeriod

The end date of a period of whole days. The period is defined according to NetworkID.

EndDatePeriod ::= DateCompact

8.25 EndDatePeriodStamp

The end date of a period of whole days. The period is defined according to NetworkID.

EndDatePeriodStamp ::= DateStamp

8.26 EndDateStamp

EndDateStamp is the final date in a period of days, e.g. it is the last day when a contract is valid. Period ends at 24 h on the end date unless an end time is explicitly specified (acceptance after this time is at the discretion of the service provider). May be used for ExpiryDate etc.

EndDateStamp ::= DateStamp

8.27 EndTime

EndTime is the end of a period of time. Acceptance after this time is at the discretion of the service provider. May be used in conjunction with EndDate or EndDateStamp where EndTime shall fall on EndDate.

EndTime ::= TimeCompact

8.28 EndTimeStamp

EndTimeStamp is the end of a period of time. Acceptance after this time is at the discretion of the service provider. May be used in conjunction with EndDate or EndDateStamp where EndTimeStamp shall fall on EndDate.

EndTimeStamp ::= TimeStamp

8.29 EntryPointer

The pointer to a directory entry.

EntryPointer ::= InstancePointer

8.30 EventClassification

EventClassification is the constructed data element describing the type of transport service to which the receipt relates. Thus it is a substructure of the event. The data element expresses both the transport mode and the action or activity (e.g. entry, exit).

EventClassification ::= SEQUENCE {

transportService TransportTypeCode,

transportServiceAction UserActionCode

}

8.31 EventDateStamp

Date of the event.

EventDateStamp ::= DateStamp

8.32 EventDisplayMessageId

Reference number stored in an event log representing the message which has been displayed.

EventDisplayMessageId ::= ReferenceNumber {255}

8.33 EventPointer

The pointer to an event.

EventPointer ::= InstancePointer

8.34 FacilityProvider

FacilityProvider is the identity of the organisation that is obliged to provide the facility specified. This can be vehicles, stops, lines etc.

FacilityProvider ::= NetworkSpecificCompanyId

8.35 FarthestPlace

The limit of range of travel measured from the place of origin for which the contract is valid.

FarthestPlace ::= LocationIdentifier

Value Assignment : a NetworkId specific value.

8.36 Fax

A fax number.

Fax ::= NetworkAccess

8.37 Forename

Forename is the forename (given name) or forenames of a person.

Forename ::= Name

8.38 HangoverPeriod

A number of days after the end of validity of a product/contract/ticket where the item should remain on the card.

HangoverPeriod ::= INT1

8.39 HolderAddress

The address of the holder.

HolderAddress ::= Address

8.40 HolderCompany

HolderCompany is the coded form of the reference to the holder if it is a company. This reference to the holder takes precedence over any other reference to holder name within the same context. HolderCompany is used when the customer party to a contract is a company rather than an individual.

HolderCompany ::= NetworkSpecificCompanyId

8.41 HolderId

The identification of a holder.

HolderId ::= ReferenceIdentifier

Value Assignment : a NetworkId specific value.

8.42 HolderProfiles

Number of holder profiles allowed, where each profile is defined by ProfileCodeIOP

HolderProfiles ::= INTS

8.43 IdentityDocumentId

The identification of an identity document of a holder.

IdentityDocumentId ::= ReferenceIdentifier

8.44 IssueDateStamp

Sale date of a contract.

IssueDateStamp ::= DateStamp

8.45 KeyVersionNumber

Gives the version number of the key used.

KeyVersionNumber ::= INT1

Value Assignment : a NetworkId specific value.

8.46 LastMinuteSale

Flag indicating that the contract has been loaded after the boarding list has been loaded into the check-in terminals.

LastMinuteSale ::= Flag

0 = no last minute sale, 1 = last minute sale

8.47 LevelIndicator

Indicates the level of nesting with in a CICO system. A value of zero (0) indicates that the contract holder is not in a CICO system, a value of one (1) indicates that the contract holder has checked in to a CICO system, a value of two (2) indicates that the contract holder has checked into two CICO systems, and so on.

LevelIndicator ::= INTS

8.48 LocationId

Data to specify a location, may contain specific and supplementary information in the format identified by LocationQualifierCode.

LocationId ::= ReferenceIdentifier

Value Assignment : a NetworkId specific value.

8.49 LocationIdentifier

An identification of a location.

LocationIdentifier ::= SEQUENCE {

memberId NetworkSpecificCompanyId,

type LocationTypeCode,

LocationId ReferenceIdentifier

}

8.50 LockTime

The number of seconds a product/contract is locked following use, during which period the product/contract shall not be used again“.

LockTime ::= INT1

8.51 MaxAbnormalEvents

The maximum number of abnormal events allowed.

MaxAbnormalEvents ::= Quantity

8.52 MostRecentPointer

Points to the most recent entry in the event log.

MostRecentPointer ::= InstancePointer

8.53 NotOKCounter

NotOKCounter is the counter at the application level maintaining a count of events that completed with status NOTOK (also called 'Abnormal Events') associated with the use of this card for this Surface Transport Application. To be incremented or reset by the Service Provider.

NotOkCounter ::= Counter

Value Assignment : Network specific.

8.54 NumberOfContracts

Number of contracts in the list of contracts.

NumberOfContracts ::= Quantity

8.55 NumberOfEntries

The total number of entries available, for example the number of records in a cyclic file.

NumberOfEntries ::= Quantity

8.56 NumberOfTimePeriods

The number of time periods in a charge period, e.g. number of weeks, days, hours

NumberOfTimePeriods ::= Quantity

8.57 PermitPeriodOfDay

This is a reference to indicate a period of the day (or periods), known by the system, where an information or contract is valid.

PermitPeridOfDay ::= PeriodOfDay

8.58 PostCodeId

The postcode of an address.

PostCodeId ::= ReferenceIdentifier

Value Assignment : a Post Office official Value.

8.59 Priority

The priority of a contract. It is used to choose a product when two or more are valid.

Priority ::= INT1

8.60 ProductOwner

Actor that specifies the pricing, usage rules and commercial rules related to a product at all points where a product is sold or used.

ProductOwner ::= NetworkSpecificCompanyId

8.61 ProductRetailer

Actor that sells a product and refunds value to a customer as authorised by a product owner.

ProductRetailer ::= NetworkSpecificCompanyId

8.62 ProductStatus

The current status of a product.

ProductStatus ::= StatusCode

8.63 ReceiptData

Free data to specify a receipt, may contain specific and supplementary information in a non-standard format.
ReceiptData ::= Databin

8.64 ReceiptPoint

The place where a receipt is issued.

ReceiptPoint ::= LocationIdentifier

Value Assignment : a NetworkId specific value.

8.65 ReservationId

The identifier of a reservation.

ReservationId ::= ReferenceIdentifier

Value Assignment : a NetworkId specific value.

8.66 RestrictedDayOfWeek

The selected day of the week on which the contract is not valid.

RestrictedDayOfWeek ::= DayOfWeek

8.67 RestrictedHalfDayOfWeek

The selected half-day of the week on which the contract is not valid.

RestrictedHalfDayOfWeek ::= HalfDayOfWeek

8.68 RestrictedLocation

A location where the contract is not valid.

RestrictedLocation ::= LocationIdentifier

Value Assignment : a NetworkId specific value.

8.69 RestrictedPeriodOfDay

This is a reference to indicate a period of the day (or periods), known by the system, where an information or contract is not valid.

RestrictedPeriodOfDay ::= PeriodOfDay

Value assignment : Network specific

8.70 RestrictionEnd

The date and end time of a restriction (e.g. a restriction in validity of a contract).

RestrictionEnd ::= DateTimeStamp

Value assignment : network specific

8.71 RestrictionEndDate

The last day of a restriction in contract validity.

RestrictionEndDate ::= DateStamp

8.72 RestrictionStart

The date and start time of a restriction (e.g. a restriction in validity of a contract).

RestrictionStart ::= DateTimeStamp

Value assignment : network specific

8.73 SalesPoint

Identifies the point of sales.

SalesPoint ::= LocationIdentifier

8.74 SecondaryFlag

Marks a data structure as relating to a secondary holder.

SecondaryFlag ::= Flag

8.75 SectionNumber

Indicates one part of a line.

SectionNumber ::= ReferenceNumber

8.76 SecurityVersion

Identifies the algorithm and key used.

```
SecurityVersion ::= SEQUENCE {
    securityIdAlgorithmId          AlgorithmId,
    securityIdKeyVersionNumber     KeyVersionNumber
}
```

8.77 SerialNumber

The serial number of an item of equipment, or an instance number of an application or product datagrouping, etc.

SerialNumber ::= ReferenceNumber

Value Assignment : a NetworkId specific value.

8.78 ServiceOperator

Actor that provides services to the customer against the use of a product or permission.

ServiceOperator ::= NetworkSpecificCompanyId

8.79 StartDate

StartDate is the first date in a period of days, e.g. it is the first day when a contract is valid. Period starts at 0 h on the start date unless a start time is explicitly specified (acceptance before this time is at the discretion of the service provider).

StartDate ::= DateCompact

8.80 StartDatePeriod

The start date of a period of whole days. The period is defined according to NetworkID.

StartDatePeriod ::= DateCompact

8.81 StartDatePeriodStamp

The start date of a period of whole days. The period is defined according to NetworkID.

StartDatePeriodStamp ::= DateStamp

8.82 StartDateStamp

StartDateStamp is the first date in a period of days, e.g. it is the first day when a contract is valid. Period starts at 0 h on the start date unless a start time is explicitly specified (acceptance before this time at the discretion of the service provider)

StartDateStamp ::= DateStamp

8.83 StartTime

StartTime is the start of a period of time. Acceptance before this time is at the discretion of the service provider. May be used in conjunction with StartDate or StartDateStamp.

StartTime ::= TimeCompact

8.84 StartTimeStamp

StartTimeStamp is the start of a period of time. Acceptance before this time is at the discretion of the service provider. May be used in conjunction with StartDate or StartDateStamp.

StartTimeStamp ::= TimeStamp

8.85 StructureReferenceNumber

Number applicable to a specific data structure.

StructureReferenceNumber ::= ReferenceNumber

8.86 Surname

Surname is the surname (family name) of a person.

Surname ::= Name

8.87 Telephone

A phone number.

Telephone ::= NetworkAccess

8.88 TestFlag

When set, indicates that the application is used for testing only.

TestFlag ::= Flag

8.89 Time

Data element to specify the time of day when a specified event will be triggered.

Time ::= TimeCompact

8.90 TransactionOperator

Actor that performs the actual transaction.

TransactionOperator ::= NetworkSpecificCompanyId

8.91 TransactionSequenceNumber

The sequence number of the operation.

TransactionSequenceNumber ::= SequenceNumber

8.92 UnblockInstanceNumber

The UnBlockInstanceNumber is concatenated with the application identity or product/contract identity to form a string that is entered into any hot card file etc. This is done so that applications or product/contracts that were previously blocked by hotlisting can be "unhotted" by incrementing the number at a properly authorised terminal. If the card is then presented to a terminal which still contains the hot list, the card will not be blocked again, because the instance number is now different.

UnblockInstanceNumber ::= SequenceNumber

8.93 UserData

Free data to specify data for user, may contain specific and supplementary information in a non-standard format.

UserData ::= Databin

8.94 ValidationCounter

The accumulative number of validations.

ValidationCounter ::= Counter

8.95 ValidationStatus

Gives the status of the validation referenced by the corresponding codelist.

ValidationStatus ::= StatusCode

8.96 ValidDayOfExpiry

When set to one (1) indicates that the contract is valid on the day of expiry.

ValidDayOfExpiry ::= Flag

8.97 ValidDayOfIssue

When set to one (1) indicates that the contract is valid on the day of issue.

ValidDayOfIssue ::= Flag

8.98 ValidityCheckFlag

This flag shall be set to one if the validity of the contract was not fully checked at the time of validation.
ValidityCheckFlag ::= Flag

8.99 ValidityDuration

The duration of validity of a contract.

ValidityDuration ::= Duration

8.100 VehicleId

String to uniquely identify a vehicle in a specific environment.

VehicleId ::= ReferenceIdentifier

8.101 VersionNumberFor1545

The version number of this standard.

VersionNumberFor1545 ::= VersionNumber (8)

Value Assignment - Octet aligned:

Using the ‘release’ and ‘revision’ coding of VersionNumber

‘10’H ENV 1545 (May 1998)

‘20’H EN 1545

‘30’H EN 1545 (2015 Revision)

9 Encoding rules

9.1 General

This clause gives the encoding rules (i.e. transfer syntax) for the ASN.1 types as defined in this European Standard.

9.2 Basic encoding rules (BER)

Data types shall be encoded according to the Basic Encoding Rules, as defined in ISO/IEC 8825-1. However, particular applications, such as Automatic Fee Collection (AFC), may choose to use another transfer syntax (PER) than BER.

9.3 Alternative encoding rules

9.3.1 General

This clause refers to encoding rules as alternatives to BER.

9.3.2 Packed encoding rules

Different versions of ASN.1 Packed Encoding Rules shall be as defined in ISO/IEC 8825-2 and can be applied to data types existing in the ASN.1 modules in Clause 8.

9.3.3 Other encoding rules

This European Standard does not exclude the use of other encoding rules for certain applications.

Those encoding rules can be either standardised or proprietary.

9.4 Value and size range definitions

Some ASN.1 types which are defined in Clauses 6, 7 and 8, are based on subtype identifiers.

This sub clause lists these identifiers but does not specify their values. These values specifications are left to the applications.

amountRange

authenticatorSize

bitMapRange

counterRange

databinSize

companyIdSize

instancePointerRange

pTagRange

quantityRange

referenceIdentifierRange

referenceNumberRange

sequenceNumberRange

signedMinValueAmountRange

signedMaxValueAmountRange

timeRealRange

versionNumberSize

10 Backwards compatibility

Backwards compatibility has been maintained with version 1.0 defined in ENV 1545-1:1998 and ENV 1545-2:1998, wherever possible. Implementers of ENV 1545-1:1998 and ENV 1545-2:1998 are advised to review their implementation for compatibility with this European Standard.

11 Transport general module definition

This module definition is used to identify the TransportGeneral module. It will be imported into one of the transport application specific modules defined in other parts of EN 1545.

The following object identifier and object descriptor values are assigned to identify and describe the ASN.1 module TransportGeneral2:

TransportGeneral2 ::= OBJECT IDENTIFIER { iso (1) identifiedorg(3) cen.std (0162) 0001 1

EN 1545 01 (01545) abstractsyntax (1) transportGeneral2 (2) }

and

"EN 1545 ASN.1 Module for TransportGeneral V2"

-- Pretty-printed by Asnp, the pretty-printer of France Telecom R&D

```
TransportGeneral2 DEFINITIONS AUTOMATIC TAGS ::=
BEGIN
EXPORTS ALL;
IMPORTS AID
FROM Iso7816-5
LicPlateNumber
FROM Iso14816;
Address ::= UTF8String(SIZE (0..255))
Amount{INTEGER:amountRange} ::= INTEGER(0..amountRange)
ApplicationInstanceNumber ::= ReferenceNumber
Authenticator ::= OCTET STRING(SIZE (1..authenticatorSize))
BCDString ::=
CHARACTER STRING
(WITH COMPONENTS {
identification (WITH COMPONENTS {
fixed PRESENT
})
})
BCDStringType ::= BCDString
BitMap{INTEGER:bitMapRange} ::= BIT STRING(SIZE (bitMapRange))
Capacity ::= Value{capacity}
CompanyId{INTEGER:companyIdSize} ::= OCTET STRING(SIZE (companyIdSize))
Counter{INTEGER:counterRange} ::= INTEGER(0..counterRange)
CountryAlpha ::= PrintableString(SIZE (3))
CountryNumeric ::= INTEGER(0..1023)
Currency ::= INTEGER(0..1023)
Databin{INTEGER:databinSize} ::= BIT STRING(SIZE (0..databinSize))
DateCompact ::= SEQUENCE {
year   BIT STRING(SIZE (7)),
month BIT STRING(SIZE (4)),
day BIT STRING(SIZE (5))}
Datef ::= SEQUENCE {
year   BCDString(SIZE (2)),
month BCDString(SIZE (1)),
day BCDString(SIZE (1))}
DateStamp ::= BIT STRING(SIZE (14))
DateTimeCompact ::= SEQUENCE {date DateCompact,
time TimeCompact}
```

```
DateTimeStamp ::= INT3
DayOfWeek ::= BIT STRING(SIZE (8))
Duration ::= Value{time}
Flag ::= BOOLEAN
HalfDayOfWeek ::= BIT STRING(SIZE (16))
HalfDayType ::= BIT STRING(SIZE (7))
IAI ::= OCTET STRING(SIZE (12))
IIN ::= BCDString(SIZE (6))
InstancePointer{INTEGER:instancePointerRange} ::=
INTEGER(0..instancePointerRange)
INT1 ::= INTEGER(0..255)
INT2 ::= INTEGER(0..65535)
INT3 ::= INTEGER(0..16777215)
INT4 ::= INTEGER(0..4294967295)
INTM ::= INTEGER(0..63) INTP ::=
INTEGER(0..31) INTS ::= INTEGER(0..15)
LanguageAlpha ::= PrintableString(SIZE (3))
LanguageId ::= CHOICE {
languageAlpha [0] LanguageAlpha,
languageCode [1] LanguageCode}
Length ::= Value{length}
MappingType ::= INTS
Value{MeasuredParameterType:param} ::= SEQUENCE {
unit MEASUREDPARAMETER{param}.&Unit({DefinedMeasuredParameters}),
value INTEGER}
Interval{MeasuredParameterType:param} ::= SEQUENCE {
lowerBound Value{param} OPTIONAL,
upperBound Value{param} OPTIONAL}
MEASUREDPARAMETER{MeasuredParameterType:param} ::= CLASS {
&measuredParameterType MeasuredParameterType(param),
&MeasuredParameterUnit }
WITH SYNTAX {&measuredParameterType &MeasuredParameterUnit}
MeasuredParameterType ::= ENUMERATED {
number(0), time(1), length(2), speed(3), weight(4), payment(5), capacity(6)
}
DefinedMeasuredParameters MEASUREDPARAMETER ::=
{{number NumberUnit} | {time TimeUnit} | {length LengthUnit} |
{speed SpeedUnit} | {weight WeightUnit} | {payment PaymentUnit} |
{capacity CapacityUnit}}
Name ::= UTF8String(SIZE (0..39))
NetworkAccess ::= SEQUENCE {
protocol  COMMUNICATIONMEDIUM.&protocol({DefinedMedia}),
service   COMMUNICATIONMEDIUM.&service({DefinedMedia}{@.protocol}),
networkAddress
COMMUNICATIONMEDIUM.&AddressType({DefinedMedia}{@.protocol}) OPTIONAL, linkedProtocol
NetworkAccess OPTIONAL}
COMMUNICATIONMEDIUM ::= CLASS {
&protocol CommunicationProtocol,
&service  CommunicationService,
&AddressType }
WITH SYNTAX {&communicationProtocol &communicationService &AddressType}
CommunicationProtocol ::= ENUMERATED {
gsm(0), ip(1), wap(2), terrestrial(3), ...
}
CommunicationService ::= ENUMERATED {sms(0), gprs(1), tcp(2), udp(3), ...
}
DefinedMedia COMMUNICATIONMEDIUM ::=
```

```
{{gsm sms PhoneNumber} | {gsm gprs PhoneNumber} |
{internetProtocol tcp IpAddress} | {internetProtocol udp IpAddress} |
{wap NULL IpAddress} | {terrestrial NULL PhoneNumber}, ...}
IpAddress ::= OCTET STRING
PhoneNumber ::= SEQUENCE OF NumericString --Including the country code
NetworkId ::= BIT STRING(SIZE (24))
NetworkSpecificCompanyId ::= SEQUENCE {network NetworkId,
company CompanyId}
Number ::= Value{number}
NumberUnit ::= INTEGER(0..255) -- Exponent of the power of 10
Payment ::= Value{payment}
PayUnitMap ::= BitMap(SIZE (4))
Percentage-0 ::= INTEGER(0..100)
Percentage-1 ::= INTEGER(0..1000)
Percentage-2 ::= INTEGER(0..10000)
PeriodOfDay ::= INT1
Permission ::= INT1
PointerValue ::= OCTET STRING(SIZE (1..16))
PTag{INTEGER:pTagRange} ::= INTEGER(0..pTagRange)
Quantity{INTEGER:quantityRange} ::= INTEGER(0..quantityRange)
ReferenceIdentifier ::= OCTET STRING(SIZE (referenceIdentifierRange))
ReferenceNumber{INTEGER:referenceNumberRange} ::=
INTEGER(0..referenceNumberRange)
Restriction ::= INT2
SequenceNumber{INTEGER:sequenceNumberRange} ::= INTEGER(0..sequenceNumberRange)
ShortName ::= UTF8String(SIZE (8))
SignedAmount{INTEGER:signedMinValueAmountRange,
INTEGER:signedMaxValueAmountRange} ::=
INTEGER(signedMinValueAmountRange..signedMaxValueAmountRange)
SignedInteger1 ::= INTEGER(-127..127)
SignedInteger2 ::= INTEGER(-32767..32767)
SignedInteger3 ::= INTEGER(-83886077..83886077)
Speed ::= Value{speed}
TimeCompact ::= BIT STRING(SIZE (16))
TimeMeasure ::= Value{time}
TimeReal{INTEGER:timeRealRange} ::= INTEGER(0..timeRealRange)
TimeStamp ::= BIT STRING(SIZE (11))
VehicleNumber ::= LicPlateNumber
VersionNumber{INTEGER:versionNumberSize} ::=
BIT STRING(SIZE (versionNumberSize))
Weight ::= Value{weight}
CapacityUnit ::= INT1
 --iso7372PackageCode7065         (0..153),
 --rfuCEN                         (154..175),
 --volumeNotFurtherSpecified      (176),
 --volumeDecilitres               (177),
 --volumeLitres                   (178),
 --volumeCubicMetres              (179),
 --volumeCountrySpecificUse       (180),
 --networkIdSpecific              (181..191),
 --livestocknotFurtherSpecified   (192),
 --rfuCEN                         (193..200),
```

```
 --livestockCountrySpecificUse    (201),
 --networkIdSpecific              (202..207),
 --countrySpecific                (208..255)
CommercialProductCode ::= INT1
-- System-fee                    (1)   Shipment-cost                                (2)
-- diesel                        (3)   diesel-bio                                   (4)
-- Red-diesel                    (5)   Petrol                                       (6)
-- Petrol-unleaded               (7)   Super                                        (8)
-- Super-unleaded                (9)   Super-plus                                   (10)
-- Two-stroke-mixture            (11   Fuel-98                                      (12)
-- Super-unleaded-power          (13   Super-plus-racing                            (14)
-- Liquified-petroleum-gas       (15   Motor-oil                                    (16)
-- Motor-oil-II                  (17   Transmission-oil                             (18)
-- Batteries                     (19   maps                                         (20)
-- Other-accessories             (21   other-accessories-red-VAT                    (22)
-- Other-accessories-red-VAT-II  (23   other-accessories-VAT-free                   (24)
-- Car-wash                      (25   vehicle-services                             (26)
-- Testing-services              (27   miscellaneous-station                        (28)
--Miscellaneous-station-red-VAT (29   miscellaneous-station-VAT-free                (30)
--Tyres                         (31   tyres VAT free                               (32)
--Tyres-delivered               (33   tyres-delivered-VAT-free                     (34)
--Tyre-fitting                  (35   tyre-fitting-VAT-free                        (36)
--Tyre-fitting-mobile           (37   tyre-fitting-mobile-VAT-free                 (38)
--Parts-tyres                   (39   parts-tyres-VAT-free                         (40)
--Repair                       (41   repair VAT free                              (42)
--miscellaneous-garage         (43   Miscellaneous-garage-red-VAT                 (44)
--miscellaneous-garage-VAT-free (45  Parts                                        (46)
--parts-red-VAT                (47   Parts-VAT-free                              (48)
--parts-sales                  (49   Parts-sales-VAT-free                        (50)
--parts-sales-red-VAT          (51   diesel-to-garage                            (52)
--towing                       (53   towing-VAT-free                             (54)
--insurance                    (55   insurance-VAT-free                          (56)
--cooling-system-service       (57   other-service                              (58)
--other-service-red-VAT        (59   other-service-VAT-free                      (60)
--labour                       (61   labour-red-VAT                             (62)
--labour-VAT-free              (63   toll                                       (64)
--toll-red-VAT                 (65   toll-VAT-free                              (66)
--tunnel-fee                   (67   tunnel-fee-red-VAT                        (68)
--tunnel-fee-VAT-free          (69   bridge-toll                               (70)
--bridge-toll-red-VAT          (71   bridge-toll-VAT-free                       (72)
--entry-fee                    (73   service-for-road-tax                      (74)
--HGV-tax                      (75   combined-traffic                          (76)
--Combined-traffic-VAT-free    (77   Euro-vignette                             (78)
--Vignette-superior-to-3,5-t   (79   Vignette-AUT-inferior-or-equal-to-12-t(80)
--Transport-accompaniment      (81   Cleaning                                   (82)
--Cleaning-VAT-free            (83   mobile-cardware                           (84)
--Mobile-hardware              (85   rental-service                            (86)
```

```
--Rental-service-red-VAT          (87   parking-area-charge                        (88)
--Ferry-charge                    (89   ferry-charge-VAT-free                      (90)
--Ferry-charges-inland-waters     (91   boarder-clearance                          (92)
--Boarder-clearance-VAT-free      (93   loan-payment-VAT-free                      (94)
--Repair-breakdown                (95   repair-breakdown-VAT-free                  (96)
--Towing-breakdown                (97   towing-breakdown-VAT-free                  (98)
--Delivery-costs                  (99   delivery-costs-VAT-free                   (100)
--Transport-insurance             (101) transport-insurance VAT-free             (102)
--Tyres-breakdown                 (103) tyres-breakdown-VAT-free                 (104)
--Replacement-vehicle             (105) replacement-vehicle-VAT-free             (106)
--Intervention                    (107) intervention-red-VAT                     (108)
--Intervention-VAT-free           (109)
--rfuCEN                          (110.. .180)
--NetworkIdSpecific               (181.. .255)
ConditionCode ::= ENUMERATED {
   not-exchangeable(0), before-departure(1), up-to-1-hour-before-departure(2), up-
   to-2-hours-before-departure(3),   up-to-1-day-before-departure(4),   up-to-2-
   days-before-departure(5),  up-to-1-week-before-departure(6),  up-to-2-weeks-
   before-departure(7),   up-to-1-month-before-departure(8),   up-to-2-months-
   before-departure(9), up-to-1-hour-after-departure(10), up-to-2-hours-after-
   departure(11),      up-to-1-day-after-departure(12),      up-to-2-days-after-
   departure(13),     up-to-1-week-after-departure(14),     up-to-2-weeks-after-
   departure(15),    up-to-1-month-after-departure(16),    up-to-2-months-after-
   departure(17), rfuCEN1(18), rfuCEN2(19), rfuCEN3(20), rfuCEN4(21), rfuCEN5(22),
   rfuCEN6(23),   rfuCEN7(24),   networkIdSpecific1(25),   networkIdSpecific2(26),
   networkIdSpecific3(27),    networkIdSpecific4(28),    networkIdSpecific5(29),
   networkIdSpecific6(30), networkIdSpecific7(31)}

DayOfValidityCode ::= INTS
--contract-is-not-valid-on-the-day-of-expiry  (0),
--contract-is-valid-on-the-day-of-expiry      (1),
--contract-is-not-valid-on-the-day-of-issue   (2),
--contract-is-valid-on-the-day-of-issue       (3),
--rfuCEN                                      (4..15)
DestinationOrOriginCode ::= ENUMERATED {origin(0), destination(1)}
DeviceTypeCode ::= INTS
--otherwise-specified                   (0),
--registrationTerminal                  (1)(used  for cico/bibo etc.
--ticketVendingMachine                  (2)
--inspectionTerminal                    (3)
--customerMediaIssuingTerminal          (4)
--customerMediaReturnTerminal           (5)
--validator                             (6)
--multiFunctionTerminal                 (7)
--informationTerminal                   (8)
--loadingTerminal                       (9)
--rfuCEN                                (10..12),
--networkIdSpecific                     (13..15)
DirectionCode ::= ENUMERATED {outward(0), inward(1)}
EntitlementTypeCode ::= ENUMERATED {
```

```
no-entitlement(0), warrant(1), limited-free-ride(2), proportional-fare(3), flat-
fare-discount(4), flat-fare(5), charge-to-account(6), subscription(7), frequent-
traveller(8),  senator(9),  premium(10),  gold-status(11),  silver-status(12),
capped-fare(13),  free-travel(14),  half-fare(15),  rfuCEN1(16),  rfuCEN2(17),
rfuCEN3(18), rfuCEN4(19), rfuCEN5(20), rfuCEN6(21), rfuCEN7(22), rfuCEN8(23),
rfuCEN9(24),          networkIdSpecific1(25),          networkIdSpecific2(26),
networkIdSpecific3(27),     networkIdSpecific4(28),     networkIdSpecific5(29),
networkIdSpecific6(30), networkIdSpecific7(31)}
EventTypeCode ::= ENUMERATED {
not-specified(0), sale(1), validation-outward-journey-if-return-ticket(2), undo-
previous-event-without-refund(3),  str-load(4),  str-autoload(5),  validation-
return-journey(6), str-debit(7), exchange(8), redeem-loyalty-points(9), undo-
previous-event-with-refund(10), check-in(11), check-out(12), activate-stored-
ticket(13),record-of-multiple-leg-journey(14), cta-payment-received(15), check-
in-transfer(16),  be-in-transfer(17),  user-modification(18),  consumed(19),
marked-as-blocked(20),      undo-blocking(21),      be-in(22),      be-out(23),
interruption(24),  refund-authorised(25),  inspection(26),  in-out(27),  undo-
validation(28),         networkIdSpecific1(29),         networkIdSpecific2(30),
networkIdSpecific3(31)}
GenderCode ::= BIT STRING(SIZE (2))
-- '00'B: not known
-- '01'B: male
-- '10'B: female
-- '11'B: unspecified
HotListStatusCode ::= ENUMERATED {
ok(0),  blocked-Undefined(1),  blocked-Stolen(2),  blocked-Lost(3),  blocked-
Refunded(4),  blocked-OverLimit(5),  monitored(6),  auto-reload-disabled(7),
unblocked(8),  blocked-misuse(9),  rfuCEN1(10),  rfuCEN2(11),  rfuCEN3(12),
networkIdSpecific1(13), networkIdSpecific2(14), networkIdSpecific3(15)}
LanguageCode ::= INT1
-- Undefined                  (0),
-- Abkhazian; Abkhaz;         ab (1),
-- Afan Oromo; Oromo; Galla   om (2),
-- Afar                       aa (3),
-- Afrikaans                  af (4),
-- Akan                       ak (5),
-- Albanian                   sq (6),
-- Amharic                    am (7),
-- Arabic                     ar (8),
-- Armenian                   hy (9),
-- Assamese                   as (10),
-- Avar; Avarish              av (11),
-- Avestan                    ae (12),
-- Aymara                     ay (13),
-- Azerbaijani                az (14),
-- Bambara                    bm (15),
-- Bashkir                    ba (16),
-- Basque                     eu (17),
```

```
-- Belarusian                  be (18),
-- Bengali; Bangla             bn (19),
-- Bhutani; Butanese;          dz (20),
Dzongkha
-- Bihari                      bh (21),
-- Bislama                     bi (22),
-- Bosnian                     bs (23),
-- Breton                      br (24),
-- Bulgarian                   bg (25),
-- Burmese; Myanmar            my (26),
-- Cambodian; Khmer            km (27),
-- Castilian; Spanish          es (28),
-- Catalan                     ca (29),
-- Chamorro                    ch (30),
-- Chechen                     ce (31),
-- Chichewa; Chewa; Nyanja     ny (32),
-- Chinese                     zh (33),
-- Chuang; Zhuang              za (34),
-- Church Slavonic; Church     cu (35),
Slavic; Old Slavonic; -- Old
Church Slavonic; Old
Bulgarian
-- Chuvash                     cv (36),
-- Cornish                     kw (37),
-- Corsican                    co (38),
-- Cree                        cr (39),
-- Croatian                    hr (40),
-- Czech                       cs (41),
-- Danish                      da (42),
-- Dutch                       ni (43),
-- English                     en (44),
-- Esperanto                   eo (45),
-- Estonian                    et (46),
-- Ewe                         ee (47),
-- Faroese; Faeroese           fo (48),
-- Fijian                      fj (49),
-- Finnish                     fi (50),
-- French                      fr (51),
-- Frisian                     fy (52),
-- Fulah; Fula; Fulani;        ff (53),
Fulfulde; Peul
```

```
-- Gaelic; Scottish Gaelic    gd (54),
-- Galician; Gallegan         gi (55),
-- Ganda; Luganda             lg (56),
-- Georgian                   ka (57),
-- German                     de (58),
-- Gikuyu; Kikuyu             ki (59),
-- Greenlandic; Kalaallisut   kl (60),
-- Guarani                    gn (61),
-- Gujarati                   gu (62),
-- Hausa                      ha (63),
-- Hebrew                     he (64),
-- Herero                     hz (65),
-- Hindi                      hi (66),
-- Hiri Motu                  ho (67),
-- Hungarian                  hu (68),
-- Icelandic                  is (69),
-- Ido                        io (70),
-- Igbo                       ig (70),
-- Indonesian                 id (72),
-- Interlingue                ie (73),
-- Irish                      ga (74),
-- Italian                    it (75),
-- Japanese                   ja (76),
-- Javanese                   jv (77),
-- Kannada                    kn (78),
-- Kanuri                     kr (79),
-- Kashmiri                   ks (80),
-- Kazakh                     kk (81),
-- Kikuyu;Gikuyu              ki (82),
-- Kinyarwanda; Rwanda        rw (83),
-- Kirundi; Rundi             rn (84),
-- Kiswahili; Swahili         sw (85),
-- Komi                       kv (86),
-- Kongo                      kg (87),
-- Korean                     ko (88),
-- Kurdish                    ku (89),
-- Kwanyama; Kuanyama         kj (90),
-- Kyrgyz; Kirghiz            ky (91),
-- Laotian; Lao               lo (92),
```

```
-- Latin                        la (93),
-- Latvian                      lv (94),
-- Lingala                      ln (95),
-- Lithuanian                   lt (96),
-- lnterlingua                  ia (97),
-- lnuktitut                    iu (98),
-- lnupiaq                      ik (99),
-- Luba-Katanga                 lu (100),
-- Luganda; Ganda               lg (101),
-- Luxembourgish                lb (102),
-- Macedonian                   mk (103),
-- Malagasy                     mg (104),
-- Malay                        ms (105),
-- Malayalam                    ml (106),
-- Maldivian; Divehi            dv (107),
-- Maltese                      mt (108),
-- Manx                         gv (109),
-- Maori                        mi (110),
-- Marathi                      mr (111),
-- Marshallese                  mh (112),
-- Modem Greek (post 1453)      el (113),
-- Moldavian                    mo (114),
-- Mongolian                    mn (115),
-- Nauruan                      na (116),
-- Navajo; Navaho               nv (117),
-- Ndonga                       ng (118),
-- Nepali                       ne (119),
-- North Ndebele                nd (120),
-- Northern Sami                se (121),
-- Norwegian                    no (122),
-- Norwegian Bokmál             nb (123),
-- Norwegian Nynorsk -          nn (124),
-- Occitan; Provençal (post     oc (125),
1500)
-- Ojibwa                       oj (126),
-- Oriya                        or (127),
-- Ossetian; Ossetic            os (128),
-- Pali                         pi (129),
-- Pashto; Pushto               ps (130),
```

```
-- Persian; Farsi             fa (131),
-- Polish                     pl (132),
-- Portuguese                 pt (133),
-- Punjabi; Panjabi -         pa (134),
-- Quechua                    qu (135),
-- Rhaeto-Romance             rm (136),
-- Romanian                   ro (137),
-- Russian                    ru (138),
-- Rwanda; Kinyarwanda        rw (139),
-- Samoan                     sm (140),
-- Sango; Sangho              sg (141),
-- Sanskrit                   sa (142),
-- Sardinian                  sc (143),
-- Serbian                    sr (144),
-- Serbo-Croatian             sh (145),
-- Sesotho; Southern Sotho    st (146),
-- Setswana; Tswana           tn (147),
-- Shona                      sn (148),
-- Sindhi                     sd (149),
-- Sinhala; Sinhalese;        si (150),
Singhalese
-- Slovak                     sk (151),
-- Slovenian                  sl (152),
-- Somali                     so (153),
-- South Ndebele              nr (154),
-- Spanish; Castilian         es (155),
-- Sundanese                  su (156),
-- Swahili; Kiswahili         sw (157),
-- Swazi; Swati; Siswati      ss (158),
-- Swedish                    sv (159),
-- Tagalog                    tl (160),
-- Tahitian                   ty (161),
-- Tajiki                     tg (162),
-- Tamil                      ta (163),
-- Tatar                      tt (164),
-- Telugu                     te (165),
-- Thai                       th (166),
-- Tibetan                    bo (167),
-- Tigrinya                   ti (168),
```

```
-- Tongan (Tonga Islands)     to (169),
-- Tsonga                     ts (170),
-- Turkish                    tr (171),
-- Turkmen                    tk (172),
-- Twi                        tw (173),
-- Uighur                     ug (174),
-- Ukrainian                  uk (175),
-- Urdu                       ur (176),
-- Uzbek                      uz (177),
-- Venda                      ve (178),
-- Vietnamese                 vi (179),
-- Volapuk                    vo (180),
-- Waltoon                    wa (181),
-- Welsh                      cy (182),
-- Wolof                      wo (183),
-- Xhosa                      xh (184),
-- Yiddish                    yi (185),
-- Yoruba                     yo (186),
-- Zulu                       zu (187),
-- RFU CEN                    (188..255)
LegislationCode ::= ENUMERATED {none(0), civ(1),
rfuCEN1 (2), rfuCEN2 (3), rfuCEN3 (4),
networkIdSpecific (5..7)}
LengthUnit ::= ENUMERATED {
milliMetre(0), centiMetre(1), deciMetre(2), metre(3), tenMetres(4), hundredMetres(5),
kiloMetre(6), miles(7)}
LocationQualifierCode ::= INT1
--alphanumericAndSpecialCharacters   (0),
--GlobalPositioningSystem  (16),
--geographicalDataFile  (32),
--RadioDataSystem   (48),
--GlobalSystemforMobileCommunication(64),

--lowOrbiter  (80),
--DedicatedShortRangeCommunication   (96),
--GNSS (112)
--Galileo (128)
--UIC  (144),
--SNCF (145),
--IATA (146),
--mutuallyAgreed (147 .. 159),
--rfuCEN  (160 .. 199),
--networkIdSpecific (200 .. 255)
LocationTypeCode ::= INT1
--busStopStation (0),
--metroStation   (1),
--railwayStation (2),
--tramStopStation   (3),
```

```
--ferryPort  (4),
--tollPlaza  (5),
--parkingGarageLotPlace (6),
--taxationStation  (7),
--purseLoadAgent (8),
--travelAgent (9),
--bankPostOffice (10),
--retailAgent (11),
--tollTunnel  (12),
--garage  (13),
--fuelStation (14),
--customs (15),
--areaID  (16),
--corridor  (17),
--virtualGantry  (18),
--stop-on-request  (19), between official stops
--areaBitMapValidInZone (20),
--areaBitMapValidToZone (21),
--taxiRank  (22),
--latitudeLongitude (23),
--latitudeLongitudeAltitude  (24),
--geographicalDataFile (25),
--UIC  (26),
--IATA (27),
--place-of-work  (28),
--school-college-university  (29),
--mixedTypeStation  (30),
--rfuCEN  (31..199),
--networkIdSpecific (200..255)
PersonalisationBiometricCode ::= INTP
--unspecified (0),
--photograph  (1),
--iris (2),
--fingerprint (3),
--dna  (4),
--facialGeometry (5),
--signature  (6),
--rfuCEN  (7..15),
--networkIdSpecific (16..31)
PersonalisationTypeCode ::= ENUMERATED {
anonymous(0), identified(1), --
information in the back office
personalised(2), networkIdSpecific(3..7)}
PointerQualifierCode ::= INT1
--masterFileLevel  (0),
--surfaceTransportApplication (1),
--selectedDedicatedFilelevel  (2),
--fullyQualifiedPathSupplied  (3),
--otherISO/IEC7816-4-5Environments  (4.. 15),
--dedicatedFileNameSupplied  (16),
--twoOctetElementaryFileIdSupplied  (17),
--otherCardSpecificEnvironments  (18..32),
--absoluteAddress  (33),
--relativeOffset (34),
--otherSequentialStorageEnvironments (35 .. 47),
--berTagSpecification  (48),
--perTagSpecification  (49),
```

```
--otherASN1Environments(50 .. 63),
--rfuCEN  (64 .. 255)
PreferenceTypeCode ::= ENUMERATED {
not-specified(0), use-CEN-ISSS-13987-2003-e-URI(1),
networkIdSpecific(2..7)}
ProfileCodeIOP ::= INTM
Unspecified                    (0)
Adult                          (1)
Child                          (2)
Student                        (3)
Pensioner                      (4)
disabledNotfurtherspecified    (5)
disabledVisuallyImpaired       (6)
disabledHearingImpaired        (7)
Unemployed                     (8)
--staff                       (9)
--military                    (10)
--resident                    (11)
--industrialOwnedHaulage      (12)
--busTransportCompany         (13)
--longDistanceTransport       (14)
--localTransport              (15)
--commuter                    (16)
--chargeableAnimal            (17)
--chargeableObject            (18)
--scholar                     (19)
--trainee                     (20)
--police                      (21)
--motorbike                   (22)
--pushbike                    (23)
--perambulator-without-child  (24)
--senior                      (25)
--family                      (26)
--driver                      (27)
--rfuCEN                      (28 .. 63)
ProfileCodeNetwork ::= INTM
--networkIDSpecific (0..63)
ReferenceTypeCode ::= ENUMERATED {
unspecified(0), contract(1), travelDossier(2), issueControl(3), rfuCEN1(4), rfuCEN2(5),
networkIdSpecific1(6), networkIdSpecific2(7)}
RestrictTimeCode ::= PeriodOfDay ResultCode ::= BIT STRING(SIZE (8))
RevocationDetailsCode ::= ENUMERATED {
revocationReleased(0), noFundsAvailable(1), paymentMeansExpired(2), securityViolation(3),
contractNotConfirmed(4), hotlisted(5), rfuCEN(6), networkIdSpecific(7)}
```

```
RoundingCode ::= INTS
--No-rounding-rule-defined (0..3),
--Round-down-to-single-currency-unit (4),
--Round-down-to- nearest-multiple-of-5-currency-units(5),
--Round-up-to-single-currency-unit   (6),
--Round-up-to- nearest-multiple-of-5-currency-units  (7),
--rfuCEN  (8..12),
--networkIdSpecific (13..15)
SecurityServicesCode ::= BIT STRING {
digitalsignature(0),  mac(1),  symmetricEncryption(2),  asymmetricEncryption(3),
biometricAuthentication(4), rfuCEN1(5), rfuCEN2(6), networkIdSpecific(7)}
SeriousnessCode ::= ENUMERATED {
none(0), information(1), warning(2), fault(3), rfuCEN1 (4), rfuCEN2 (5), rfuCEN3
(6), rfuCEN4 (7)}
SpeedUnit ::= ENUMERATED {
metrespersecond(0), kilometresperhour(1)}
StatusCode ::= INTEGER(0..127)
--never-validated   (0),
--used-once  (1),
--validated  (2),
--contract-already-renewed (3),
--punched (4),
--cancelled  (5),
--interrupted(6),
--ok   (7),
--new  (8),
--released   (9),
--corrupted  (10),
--terminated (11),
--expired (12),
--not-available-for-validation   (13),
--free-entry (may be re-used) (14),
--active  (15),
--pre-allocated (as yet unused)  (16),
--completed-and-can-be-removed   (17),
--completed-and-cannot-be-removed(18),
--blocked (19),
--data-group-encrypted-flag   (20), (to maintain operator data privacy)
--data-group-anonymous-flag   (21), (to support anonymity requirements)
--rfuCEN  (22..32),
--pending (33),
--rfuCEN  (34..62),
--suspended  (63),
--rfuCEN  (64..87),
--disabled   (88),
--networkIdSpecific (89..124),
--suspended-contract   (125),
--invalid-contract  (126),
--invalid-and-reimbursed-contract(127)
TimeUnit ::= ENUMERATED {
millisecond(0), second(1), minute(2), hour(3), day(4), --intercalendary day not counted
week(5), month(6), year(7)}
TransactionModeCode ::= BIT STRING(SIZE (2))
TransportTypeCode ::= INTP
--not-specified-further(0),
--urbanBus   (1),
--interurbanBus  (2),
```

```
--lightTrainMassTransit (3), (Underground, Metro, LTR)
--tram (4),
--commuterTrain  (5),
--waterborneVehicle (6),
--toll (7),
--parking (8),
--taxi (9),
--highSpeedTrain (10),
--ruralBus    (11),
--expressCommuterTrain  (12),
--paraTransit (13),
--self-drive-vehicle    (14),
--coach    (15),
--locomotive  (16),
--poweredMotorVehicle   (17),
--trailer (18),
--regionalTrain  (19),
--interCity   (20),
--funicularTrain (21),
--cableCar    (22),
--selfServiceBicycle    (23),
--carSharing  (24),
--carPooling  (25),
--rfuCEN   (26...27),
--networkIdSpecific (29...31)
UserActionCode ::= INTP
--notSpecifiedFurther   (0),
--entry    (1),
--exit (2),
--passage (3),
--checkpointInspection  (4),
--autonomous  (5),
--interchange (6),
--validation  (7),
--presenceDetected   (8),
--rfuCEN   (9...19),
--networkIdSpecific (20...31)
WeightUnit ::= ENUMERATED {
gramme(0), kilogramme(1), tenKilogrammes(2), hundredKilogrammes(3), tonne(4),
rfuCEN1 (5), rfuCEN2 (6), rfuCEN3 (7)}
UserMediaTypeCode ::= INTM
   --unspecified    (0),
   --ID-1-card-ISO-IEC-7810   (1),
   --mobilePhone    (2),
   --rfuCEN         (3..31),
   --networkIdSpecific (32..63)
SecurityAlgorithmCode ::= ENUMERATED {
   not-specified (0), RSA (1), undefined-asymmetric-algorithm (2),
rfuCEN1 (3), AES (4), single-DES (5), tripel-DES-(1key) (6), triple-DES-
(2key) (7), triple-DES-(3key) (8), SHA-1 (9), MD5 (10), undefined-mac-
algorithm (11), rfuCEN2 (12), undefined-symmetric-algorithm (13),
rfuCEN3 (14), elliptic-curve-Diffie-Hellmann (15), elliptic-curve-digital-
signature-algorithm (16),
elliptic-curve-integrated-encryption-scheme (17),
   undefined-elliptic-curve-algorithm (18), rfuCEN4 (19),
networkIdSpecific   (20..31)}
AccountingId ::= ReferenceIdentifier
```

```
ActionListSequenceNumber ::= SequenceNumber
AlgorithmId ::= INT1 ApplicationId ::= AID ApplicationOwner ::=
NetworkSpecificCompanyId
BirthDate ::= Datef BirthName ::= Name BirthPlace ::= Name
CollectionAndForwardingOperator ::= NetworkSpecificCompanyId
CompanyName ::= Name ContractDependencyPointer ::= InstancePointer
ContractTypesAllowed ::= ReferenceNumber
CustomerContractProvider ::= NetworkSpecificCompanyId
CustomerNumber ::= ReferenceNumber
Date ::= DateCompact DateTime ::= DateTimeCompact
DateTimeBand ::= ReferenceNumber
DeductionPercentage ::= Percentage-0
DelayCounter ::= Counter DeviceId ::= ReferenceIdentifier
DisplayMessageNumber ::= ReferenceNumber
EmailAddress ::= NetworkAccess EndDate ::= DateCompact
EndDatePeriod ::= DateCompact EndDatePeriodStamp ::= DateStamp EndDateStamp ::=
DateStamp
EndTime ::= TimeCompact
EndTimeStamp ::= TimeStamp
EntryPointer ::= InstancePointer
EventClassification ::= SEQUENCE { transportService  TransportTypeCode,
transportServiceAction UserActionCode}
EventDateStamp ::= DateStamp EventDisplayMessageId ::= ReferenceNumber{255}
EventPointer ::= InstancePointer FacilityProvider ::= NetworkSpecificCompanyId
FarthestPlace ::= LocationIdentifier
Fax ::= NetworkAccess Forename ::= Name
HangoverPeriod ::= INT1 HolderAddress ::= Address
HolderCompany ::= NetworkSpecificCompanyId HolderId ::= ReferenceIdentifier
HolderProfiles ::= INTS
IdentityDocumentId ::= ReferenceIdentifier IssueDateStamp ::= DateStamp
KeyVersionNumber ::= INT1
LastMinuteSale ::= Flag LevelIndicator ::= INTS
LocationId ::= ReferenceIdentifier
LocationIdentifier ::= SEQUENCE {
memberId  NetworkSpecificCompanyId,
type   LocationTypeCode,
LocationId ReferenceIdentifier}
LockTime ::= INT1
MaxAbnormalEvents ::= Quantity MostRecentPointer ::= InstancePointer
NotOkCounter ::= Counter
NumberOfContracts ::= Quantity
NumberOfEntries ::= Quantity NumberOfTimePeriods ::= Quantity PermitPeridOfDay
::= PeriodOfDay
PostCodeId ::= ReferenceIdentifier
Priority ::= INT1
ProductOwner ::= NetworkSpecificCompanyId ProductRetailer ::=
NetworkSpecificCompanyId ProductStatus ::= StatusCode
ReceiptData ::= Databin
ReceiptPoint ::= LocationIdentifier ReservationId ::= ReferenceIdentifier
RestrictedDayOfWeek ::= DayOfWeek RestrictedHalfDayOfWeek ::= HalfDayOfWeek
RestrictedLocation ::= LocationIdentifier RestrictedPeriodOfDay ::= PeriodOfDay
RestrictionEnd ::= DateTimeStamp RestrictionEndDate ::= DateStamp
RestrictionStart ::= DateTimeStamp
SalesPoint ::= LocationIdentifier
SecondaryFlag ::= Flag
SectionNumber ::= ReferenceNumber
```

```
SecurityVersion ::= SEQUENCE { securityIdAlgorithmId AlgorithmId,
securityIdKeyVersionNumber KeyVersionNumber}
SerialNumber ::= ReferenceNumber ServiceOperator ::= NetworkSpecificCompanyId
StartDate ::= DateCompact StartDatePeriod ::= DateCompact StartDatePeriodStamp
::= DateStamp StartDateStamp ::= DateStamp StartTime ::= TimeCompact
StartTimeStamp ::= TimeStamp
StructureReferenceNumber ::= ReferenceNumber Surname ::= Name
Telephone ::= NetworkAccess TestFlag ::= Flag
Time ::= TimeCompact
TransactionOperator ::= NetworkSpecificCompanyId TransactionSequenceNumber ::=
SequenceNumber UnblockInstanceNumber ::= SequenceNumber UserData ::= Databin
ValidationCounter ::= Counter ValidationStatus ::= StatusCode ValidDayOfExpiry
::= Flag ValidDayOfIssue ::= Flag ValidityCheckFlag ::= Flag ValidityDuration
::= Duration VehicleId ::= ReferenceIdentifier
VersionNumberFor1545 ::= VersionNumber(8)
END
```

Annex A
(normative)

Assignment of object identifiers

```
BCDAbstractSyntaxId OBJECT IDENTIFIER ::= {
    EN1545 abstractsyntax(1) BCDString(0) }
BCDTransferSyntaxId OBJECT IDENTIFIER ::= {
    EN1545 transfersyntax(2) BCDString(0) }
EN1545 OBJECT IDENTIFIER ::= {
    iso(1) identifiedorg(3) cen.std(0162) 0001 0 EN1545(01545) }
EN1545-1 OBJECT IDENTIFIER ::= {
    iso(1) identifiedorg(3) cen.std(0162) 0001 1 EN1545 01(01545) }
TransportGeneral2 OBJECT IDENTIFIER ::= {
EN1545-1 abstractsyntax(1) transportGeneral2 (2) }
```

Annex B
(normative)

Tags

The following assignments of tags are provided.

Reference	Element or type	Tag #	Tag spec.	Tag encoding
6-1	Address	0	[APPLICATION 0]	40
6-2	Amount	1	[APPLICATION 1]	41
6-3	ApplicationInstanceNumber	2	[APPLICATION 2]	42
6-4	Authenticator	3	[APPLICATION 3]	43
6-5	BCDString	4	[APPLICATION 4]	44
6-6	BCDStringType	5	[APPLICATION 5]	45
6-7	BitMap	6	[APPLICATION 6]	46
6-8	Capacity	7	[APPLICATION 7]	47
6-9	CompanyId	8	[APPLICATION 8]	48
6-10	Counter	9	[APPLICATION 9]	49
6-11	CountryAlpha	10	[APPLICATION 10]	4A
6-12	CountryNumeric	11	[APPLICATION 11]	4B
6-13	Currency	12	[APPLICATION 12]	4C
6-14	Databin	13	[APPLICATION 13]	4D
6-15	DateCompact	14	[APPLICATION 14]	4E
6-16	Datef	15	[APPLICATION 15]	4F
6-17	DateStamp	16	[APPLICATION 16]	50
6-18	DateTimeCompact	17	[APPLICATION 17]	51
6-19	DateTimeStamp	18	[APPLICATION 18]	52
6-20	DayOfWeek	19	[APPLICATION 19]	53
6-21	Duration	20	[APPLICATION 20]	54
6-22	Flag	21	[APPLICATION 21]	55
6-23	HalfDayOfWeek	22	[APPLICATION 22]	56
6-24	HalfDayType	23	[APPLICATION 23]	57
6-25	IAI	24	[APPLICATION 24]	58
6-26	IIN	25	[APPLICATION 25]	59
6-27	INT1	26	[APPLICATION 26]	5A
6-28	INT2	27	[APPLICATION 27]	5B
6-29	INT3	28	[APPLICATION 28]	5C
6-30	INT4	29	[APPLICATION 29]	5D
6-31	INTM	30	[APPLICATION 30]	5E
6-32	INTP	31	[APPLICATION 31]	5F 1F
6-33	INTS	32	[APPLICATION 32]	5F 20
6-34	InstancePointer	33	[APPLICATION 33]	5F 21
6-35	LanguageAlpha	34	[APPLICATION 34]	5F 22
6-36	LanguageId	35	[APPLICATION 35]	5F 23
6-37	Length	36	[APPLICATION 36]	5F 24

6-38	MappingType	37	[APPLICATION 37]	5F 25
6-39	Value	38	[APPLICATION 38]	5F 26
6-39	Interval	39	[APPLICATION 39]	5F 27
6-39	MEASUREDPARAMETER	40	[APPLICATION 40]	5F 28
6-39	MeasuredParameterType	41	[APPLICATION 41]	5F 29
6-39	DefinedMeasuredParameters	42	[APPLICATION 42]	5F 2A
6-40	Name	43	[APPLICATION 43]	5F 2B
6-41	NetworkAccess	44	[APPLICATION 44]	5F 2C
6-41	COMMUNICATIONMEDIUM	45	[APPLICATION 45]	5F 2D
6-41	CommunicationProtocol	46	[APPLICATION 46]	5F 2E
6-41	CommunicationService	47	[APPLICATION 47]	5F 2F
6-41	DefinedMedia	48	[APPLICATION 48]	5F 30
6-41	IpAddress	49	[APPLICATION 49]	5F 31
6-41	PhoneNumber	50	[APPLICATION 50]	5F 32
6-42	NetworkId	51	[APPLICATION 51]	5F 33
6-43	NetworkSpecificCompanyId	52	[APPLICATION 52]	5F 34
6-44	Number	53	[APPLICATION 53]	5F 35
6-45	NumberUnit	54	[APPLICATION 54]	5F 36
6-46	ObjectIdentifier	55	[APPLICATION 55]	5F 37
6-47	Payment	56	[APPLICATION 56]	5F 38
6-48	PayUnitMap	57	[APPLICATION 57]	5F 39
6-49	Percentage-0	58	[APPLICATION 58]	5F 3A
6-50	Percentage-1	59	[APPLICATION 59]	5F 3B
6-51	Percentage-2	60	[APPLICATION 60]	5F 3C
6-52	PeriodOfDay	61	[APPLICATION 61]	5F 3D
6-53	Permission	62	[APPLICATION 62]	5F 3E
6-54	PointerValue	63	[APPLICATION 63]	5F 3F
6-55	PTag	64	[APPLICATION 64]	5F 40
6-56	Quantity	65	[APPLICATION 65]	5F 41
6-57	ReferenceIdentifier	66	[APPLICATION 66]	5F 42
6-58	ReferenceNumber	67	[APPLICATION 67]	5F 43
6-59	Restriction	68	[APPLICATION 68]	5F 44
6-60	SequenceNumber	69	[APPLICATION 69]	5F 45
6-61	ShortName	70	[APPLICATION 70]	5F 46
6-62	SignedAmount	71	[APPLICATION 71]	5F 47
6-63	SignedInteger1	72	[APPLICATION 72]	5F 48
6-64	SignedInteger2	73	[APPLICATION 73]	5F 49
6-65	SignedInteger3	74	[APPLICATION 74]	5F 4A
6-66	Speed	75	[APPLICATION 75]	5F 4B
6-67	TimeCompact	76	[APPLICATION 76]	5F 4C
6-68	TimeMeasure	77	[APPLICATION 77]	5F 4D
6-69	TimeReal	78	[APPLICATION 78]	5F 4E
6-70	TimeStamp	79	[APPLICATION 79]	5F 4F
6-71	VehicleNumber	80	[APPLICATION 80]	5F 50
6-72	VersionNumber	81	[APPLICATION 81]	5F 51
6-73	Weight	82	[APPLICATION 82]	5F 52
7-1	CapacityUnit	83	[APPLICATION 83]	5F 53

7-2	CommercialTransportProductCode	84	[APPLICATION 84]	5F 54
7-3	ConditionCode	85	[APPLICATION 85]	5F 55
7-4	DayOfValidityCode	86	[APPLICATION 86]	5F 56
7-5	DestinationOrOriginCode	87	[APPLICATION 87]	5F 57
7-6	DeviceTypeCode	88	[APPLICATION 88]	5F 58
7-7	DirectionCode	89	[APPLICATION 89]	5F 59
7-8	EntitlementTypeCode	90	[APPLICATION 90]	5F 5A
7-9	EventTypeCode	91	[APPLICATION 91]	5F 5B
7-10	GenderCode	92	[APPLICATION 92]	5F 5C
7-11	HotListStatusCode	93	[APPLICATION 93]	5F 5D
7-12	LanguageCode	94	[APPLICATION 94]	5F 5E
7-13	LegislationCode	95	[APPLICATION 95]	5F 5F
7-14	LengthUnit	96	[APPLICATION 96]	5F 60
7-15	LocationQualifierCode	97	[APPLICATION 97]	5F 61
7-16	LocationTypeCode	98	[APPLICATION 98]	5F 62
7-17	PersonalisationTypeCode	99	[APPLICATION 99]	5F 63
7-18	PersonalisationBiometricCode	100	[APPLICATION 100]	5F 64
7-19	PointerQualifierCode	101	[APPLICATION 101]	5F 65
7-20	PreferenceTypeCode	102	[APPLICATION 102]	5F 66
7-21	ProfileCodeIOP	103	[APPLICATION 103]	5F 67
7-22	ProfileCodeNetwork	104	[APPLICATION 104]	5F 68
7-23	ReferenceTypeCode	105	[APPLICATION 105]	5F 69
7-24	RestrictTimeCode	106	[APPLICATION 106]	5F 6A
7-25	ResultCode	107	[APPLICATION 107]	5F 6B
7-26	RevocationDetailsCode	108	[APPLICATION 108]	5F 6C
7-27	RoundingCode	109	[APPLICATION 109]	5F 6D
7-28	SecurityServicesCode	110	[APPLICATION 110]	5F 6E
7-29	SeriousnessCode	111	[APPLICATION 111]	5F 6F
7-30	SpeedUnit	112	[APPLICATION 112]	5F 70
7-31	StatusCode	113	[APPLICATION 113]	5F 71
7-32	TimeUnit	114	[APPLICATION 114]	5F 72
7-33	TransactionModeCode	115	[APPLICATION 115]	5F 73
7-34	TransportTypeCode	116	[APPLICATION 116]	5F 74
7-35	UserActionCode	117	[APPLICATION 117]	5F 75
7-36	WeightUnit	118	[APPLICATION 118]	5F 76
8-1	AccountingId	119	[APPLICATION 119]	5F 77
8-2	ActionListSequenceNumber	120	[APPLICATION 120]	5F 78
8-3	AlgorithmId	121	[APPLICATION 121]	5F 79
8-4	ApplicationId	122	[APPLICATION 122]	5F 7A
8-5	ApplicationOwner	123	[APPLICATION 123]	5F 7B
8-6	BirthDate	124	[APPLICATION 124]	5F 7C
8-7	BirthName	125	[APPLICATION 125]	5F 7D
8-8	BirthPlace	126	[APPLICATION 126]	5F 7E
8-9	CollectionAndForwardingOperator	127	[APPLICATION 127]	5F 7F
8-10	CompanyName	128	[APPLICATION 128]	5F 81 00
8-11	ContractDependencyPointer	129	[APPLICATION 129]	5F 81 01
8-12	ContractTypesAllowed	130	[APPLICATION 130]	5F 81 02

8-13	CustomerContractProvider	131	[APPLICATION 131]	5F 81 03
8-14	CustomerNumber	132	[APPLICATION 132]	5F 81 04
8-15	Date	133	[APPLICATION 133]	5F 81 05
8-16	DateTime	134	[APPLICATION 134]	5F 81 06
8-17	DateTimeBand	135	[APPLICATION 135]	5F 81 07
8-18	DeductionPercentage	136	[APPLICATION 136]	5F 81 08
8-19	DelayCounter	137	[APPLICATION 137]	5F 81 09
8-20	DeviceId	138	[APPLICATION 138]	5F 81 0A
8-21	DisplayMessageNumber	139	[APPLICATION 139]	5F 81 0B
8-22	EmailAddress	140	[APPLICATION 140]	5F 81 0C
8-23	EndDate	141	[APPLICATION 141]	5F 81 0D
8-24	EndDatePeriod	142	[APPLICATION 142]	5F 81 0E
8-25	EndDatePeriodStamp	143	[APPLICATION 143]	5F 81 0F
8-26	EndDateStamp	144	[APPLICATION 144]	5F 81 10
8-27	EndTime	145	[APPLICATION 145]	5F 81 11
8-28	EndTimeStamp	146	[APPLICATION 146]	5F 81 12
8-29	EntryPointer	147	[APPLICATION 147]	5F 81 13
8-30	EventClassification	148	[APPLICATION 148]	5F 81 14
8-31	EventDateStamp	149	[APPLICATION 149]	5F 81 15
8-32	EventDisplayData	150	[APPLICATION 150]	5F 81 16
8-33	EventPointer	151	[APPLICATION 151]	5F 81 17
8-34	FacilityProvider	152	[APPLICATION 152]	5F 81 18
8-35	FarthestPlace	153	[APPLICATION 153]	5F 81 19
8-36	Fax	154	[APPLICATION 154]	5F 81 1A
8-37	Forename	155	[APPLICATION 155]	5F 81 1B
8-38	HangoverPeriod	156	[APPLICATION 156]	5F 81 1C
8-39	HolderAddress	157	[APPLICATION 157]	5F 81 1D
8-40	HolderCompany	158	[APPLICATION 158]	5F 81 1E
8-41	HolderProfiles	159	[APPLICATION 159]	5F 81 1F
8-42	HolderId	160	[APPLICATION 160]	5F 81 20
8-43	IdentityDocumentId	161	[APPLICATION 161]	5F 81 21
8-44	IssueDateStamp	162	[APPLICATION 162]	5F 81 22
8-45	KeyVersionNumber	163	[APPLICATION 163]	5F 81 23
8-46	LastMinuteSale	164	[APPLICATION 164]	5F 81 24
8-47	LevelIndicator	165	[APPLICATION 165]	5F 81 25
8-48	LocationIdentifier	166	[APPLICATION 166]	5F 81 26
8-49	LocationId	167	[APPLICATION 167]	5F 81 27
8-50	LockTime	168	[APPLICATION 168]	5F 81 28
8-51	MaxAbnormalEvents	169	[APPLICATION 169]	5F 81 29
8-52	MostRecentPointer	170	[APPLICATION 170]	5F 81 2A
8-53	NotOKCounter	171	[APPLICATION 171]	5F 81 2B
8-54	NumberOfContracts	172	[APPLICATION 172]	5F 81 2C
8-55	NumberOfEntries	173	[APPLICATION 173]	5F 81 2D
8-56	NumberOfTimePeriods	174	[APPLICATION 174]	5F 81 2E
8-57	PermitPeriodOfDay	175	[APPLICATION 175]	5F 81 2F
8-58	PostCodeId	176	[APPLICATION 176]	5F 81 30
8-59	Priority	177	[APPLICATION 177]	5F 81 31

8-60	ProductOwner	178	[APPLICATION 178]	5F 81 32
8-61	ProductRetailer	179	[APPLICATION 179]	5F 81 33
8-62	ProductStatus	180	[APPLICATION 180]	5F 81 34
8-63	ReceiptData	181	[APPLICATION 181]	5F 81 35
8-64	ReceiptPoint	182	[APPLICATION 182]	5F 81 36
8-65	ReservationId	183	[APPLICATION 183]	5F 81 37
8-66	RestrictedDayOfWeek	184	[APPLICATION 184]	5F 81 38
8-67	RestrictedHalfDayOfWeek	185	[APPLICATION 185]	5F 81 39
8-68	RestrictionEnd	186	[APPLICATION 186]	5F 81 3A
8-69	RestrictionEndDate	187	[APPLICATION 187]	5F 81 3B
8-70	RestrictionStart	188	[APPLICATION 188]	5F 81 3C
8-71	RestrictedLocation	189	[APPLICATION 189]	5F 81 3D
8-72	RestrictedPeriodOfDay	190	[APPLICATION 190]	5F 81 3E
8-73	SalesPoint	191	[APPLICATION 191]	5F 81 3F
8-74	SecondaryFlag	192	[APPLICATION 192]	5F 81 40
8-75	SectionNumber	193	[APPLICATION 193]	5F 81 41
8-76	SecurityVersion	194	[APPLICATION 194]	5F 81 42
8-77	SerialNumber	195	[APPLICATION 195]	5F 81 43
8-78	ServiceOperator	196	[APPLICATION 196]	5F 81 44
8-79	StartDate	197	[APPLICATION 197]	5F 81 45
8-80	StartDatePeriod	198	[APPLICATION 198]	5F 81 46
8-81	StartDatePeriodStamp	199	[APPLICATION 199]	5F 81 47
8-82	StartDateStamp	200	[APPLICATION 200]	5F 81 48
8-83	StartTime	201	[APPLICATION 201]	5F 81 49
8-84	StartTimeStamp	202	[APPLICATION 202]	5F 81 4A
8-85	StructureReferenceNumber	203	[APPLICATION 203]	5F 81 4B
8-86	Surname	204	[APPLICATION 204]	5F 81 4C
8-87	Telephone	205	[APPLICATION 205]	5F 81 4D
8-88	TestFlag	206	[APPLICATION 206]	5F 81 4E
8-89	Time	207	[APPLICATION 207]	5F 81 4F
8-90	TransactionOperator	208	[APPLICATION 208]	5F 81 50
8-91	TransactionSequenceNumber	209	[APPLICATION 209]	5F 81 51
8-92	UnblockInstanceNumber	210	[APPLICATION 210]	5F 81 52
8-93	UserData	211	[APPLICATION 211]	5F 81 53
8-94	ValidationCounter	212	[APPLICATION 212]	5F 81 54
8-95	ValidationStatus	213	[APPLICATION 213]	5F 81 55
8-96	ValidDayOfExpiry	214	[APPLICATION 214]	5F 81 56
8-97	ValidDayOfIssue	215	[APPLICATION 215]	5F 81 57
8-98	ValidityCheckFlag	216	[APPLICATION 216]	5F 81 58
8-99	ValidityDuration	217	[APPLICATION 217]	5F 81 59
8-100	VehicleId	218	[APPLICATION 218]	5F 81 5A
8-101	VersionNumberFor1545	219	[APPLICATION 219]	5F 81 5B
7-37	UserMediaTypeCode	220	[APPLICATION 220]	5F 815C
7-38	SecurityAlgorithmCode	221	[APPLICATION 221]	5F 815D
NOTEAll tags are chosen in application and are primitive.				

Annex C
(informative)

Index

W

Bibliography

General documents

[1] EN 726-1:1994, *Identification card systems - Telecommunications integrated circuit(s) cards and terminals - Part 1: Systems overview*

[2] EN 726-2, *Identification card systems - Telecommunications integrated circuit(s) cards and terminals - Part 2: Security framework*

[3] EN 726-3:1994, *Identification card systems - Telecommunications integrated circuit(s) cards and terminals - Part 3: Application independent card requirements*

[4] EN 726-4:1994, *Identification card systems - Telecommunications integrated circuit(s) cards and terminals - Part 4: Application independent card related terminal requirements*

[5] ISO/IEC 10536-1:2000, *Identification cards — Contactless integrated circuit(s) cards — Close-coupled cards — Part 1: Physical characteristics*

[6] ISO/IEC 7816-3, *Identification cards — Integrated circuit cards — Part 3: Cards with contacts — Electrical interface and transmission protocols*

[7] ISO/IEC 8824-2, *Information technology — Abstract Syntax Notation One (ASN.1): Information object specification — Part 2*

[8] ISO/IEC 8824-3, *Information technology — Abstract Syntax Notation One (ASN.1): Constraint specification — Part 3*

[9] ISO/IEC 8824-4, *Information technology — Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications — Part 4*

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

[11] ISO/IEC 14662, *Information technology — Open-edi reference model*

[12] UIC 920-16, *Directory of code lists for the passenger domain* C 14662, *Information technology – Open-EDI reference model*

Payment related documents

[13] ETSI 300 045-1:1992, *European digital cellular telecommunications system (phase 1); Subscriber Identity Module - Mobile equipment (SIM-ME) interface specification; Part 1: Generic (GSM 11.11)*

License related documents

[14] ISO 2534:1998, *Road vehicles — Engine test code — Gross power*

[15] ISO 4100:1980, *Road vehicles — World parts manufacturer identifier (WPMI) code*

[16] ISO 6726:1988, *Mopeds and motorcycles with two wheels — Masses — Vocabulary*

This page deliberately left blank

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

bsi.

...making excellence a habit.™