

Identification card systems — Surface transport applications —

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

The European Standard EN 1545-1:2005 has the status of a
British Standard

ICS 35.240.15

National foreword

This British Standard is the official English language version of EN 1545-1:2005. It supersedes DD ENV 1545-1:1998 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/17, Cards and personal identification, which has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international/European committee any enquiries on the interpretation, or proposals for change, and keep UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

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

Cross-references

The British Standards which implement international or European publications referred to in this document may be found in the *BSI Catalogue* under the section entitled “International Standards Correspondence Index”, or by using the “Search” facility of the *BSI Electronic Catalogue* or of British Standards Online.

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

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, the EN title page, pages 2 to 94, an inside back cover and a back cover.

The BSI copyright notice displayed in this document indicates when the document was last issued.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 6 December 2005

© BSI 6 December 2005

ISBN 0 580 46907 7

Amendments issued since publication

Amd. No.	Date	Comments

English Version

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

Système 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: Datentypen, Codelisten
und generelle Datenelemente

This European Standard was approved by CEN on 19 May 2005.

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 Central Secretariat 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 Central Secretariat has the same status as the official versions.

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



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

Management Centre: rue de Stassart, 36 B-1050 Brussels

Contents

Page

1	Scope	10
2	Normative references	11
3	Terms and definitions	11
4	Abbreviations	13
5	Approach for definition of data types and data elements	13
5.1	Data types and data elements	13
5.2	ASN.1 type naming conventions	14
5.3	Existing standards	14
5.4	Value range identifiers	14
5.5	Size constraints	14
6	Elementary data types	14
6.1	Address	14
6.2	Amount	14
6.3	ApplicationInstanceNumber	15
6.4	Authenticator	15
6.5	BCDStringType	15
6.6	BitMap	15
6.7	Capacity	16
6.8	CompanyId	16
6.9	Counter	16
6.10	CountryAlpha	16
6.11	CountryNumeric	16
6.12	Currency	16
6.13	Databin	16
6.14	DateCompact	16
6.15	Datef	17
6.16	DateStamp	17
6.17	DateTimeCompact	17
6.18	DateTimeStamp	17
6.19	DayOfWeek	18
6.20	Duration	18
6.21	Flag	18
6.22	HalfDayOfWeek	18
6.23	HalfDayType	19
6.24	IAI	20
6.25	IIN	20
6.26	InstancePointer	20
6.27	INT1	20
6.28	INT2	20
6.29	INT3	21
6.30	INT4	21
6.31	INTM	21
6.32	INTP	21
6.33	INTS	21
6.34	LanguageAlpha	21
6.35	Languageld	21
6.36	Length	21
6.37	MappingType	22
6.38	MeasuredParameters	22

6.39	Name	22
6.40	NetworkAccess	23
6.41	NetworkId	24
6.42	NetworkSpecificCompanyId	24
6.43	Number	24
6.44	NumberUnit	24
6.45	ObjectIdentifier	24
6.46	Payment	24
6.47	PayUnitMap	24
6.48	Percentage-0	25
6.49	Percentage-1	25
6.50	Percentage-2	25
6.51	PeriodOfDay	25
6.52	Permission	26
6.53	PointerValue	26
6.54	Ptag	26
6.55	Quantity	26
6.56	ReferenceIdentifier	26
6.57	ReferenceNumber	26
6.58	Restriction	26
6.59	SequenceNumber	26
6.60	ShortName	27
6.61	SignedAmount	27
6.62	SignedInteger1	27
6.63	SignedInteger2	27
6.64	SignedInteger3	27
6.65	Speed	27
6.66	TimeCompact	27
6.67	TimeMeasure	28
6.68	TimeReal	28
6.69	TimeStamp	28
6.70	VehicleNumber	28
6.71	VersionNumber	28
6.72	Weight	28
7	Data elements with associated code lists	29
7.1	CapacityUnit	29
7.2	CommercialTransportProductCode	29
7.3	ConditionCode	30
7.4	DayOfValidityCode	31
7.5	DestinationOrOriginCode	31
7.6	DeviceTypeCode	31
7.7	DirectionCode	31
7.8	EntitlementTypeCode	32
7.9	EventTypeCode	32
7.10	GenderCode	33
7.11	HotListStatusCode	33
7.12	LanguageCode	34
7.13	LegislationCode	38
7.14	LengthUnit	38
7.15	LocationQualifierCode	38
7.16	LocationTypeCode	38
7.17	PersonalisationBiometricCode	39
7.18	PersonalisationTypeCode	39
7.19	PointerQualifierCode	39
7.20	PreferenceTypeCode	40
7.21	ProfileCodeOP	40
7.22	ProfileCodeNetwork	41
7.23	ReferenceTypeCode	41
7.24	RestrictTimeCode	41

7.25	ResultCode	42
7.26	RevocationDetailsCode	42
7.27	RoundingCode	42
7.28	SecurityServicesCode	43
7.29	SeriousnessCode	43
7.30	SpeedUnit	43
7.31	StatusCode	43
7.32	TimeUnit	44
7.33	TransactionModeCode	44
7.34	TransportTypeCode	44
7.35	UserActionCode	45
7.36	WeightUnit	46
8	General data elements	46
8.1	AccountingId	46
8.2	ActionListSequenceNumber	46
8.3	AlgorithmId	46
8.4	ApplicationId	46
8.5	ApplicationOwner	46
8.6	BirthDate	47
8.7	BirthName	47
8.8	BirthPlace	47
8.9	CollectionAndForwardingOperator	47
8.10	CompanyName	47
8.11	ContractDependencyPointer	47
8.12	ContractTypesAllowed	47
8.13	CustomerContractProvider	48
8.14	CustomerNumber	48
8.15	Date	48
8.16	DateTime	48
8.17	DateTimeBand	48
8.18	DeductionPercentage	48
8.19	DelayCounter	48
8.20	DeviceId	48
8.21	DisplayMessageNumber	49
8.22	EmailAddress	49
8.23	EndDate	49
8.24	EndDatePeriod	49
8.25	EndDatePeriodStamp	49
8.26	EndDateStamp	49
8.27	EndTime	49
8.28	EndTimeStamp	50
8.29	EntryPointer	50
8.30	EventClassification	50
8.31	EventDateStamp	50
8.32	EventDisplayMessageId	50
8.33	EventPointer	50
8.34	FacilityProvider	51
8.35	FarthestPlace	51
8.36	Fax	51
8.37	Forename	51
8.38	HangoverPeriod	51
8.39	HolderAddress	51
8.40	HolderCompany	51
8.41	HolderId	52
8.42	HolderProfiles	52
8.43	IdentityDocumentId	52
8.44	IssueDateStamp	52
8.45	KeyVersionNumber	52
8.46	LastMinuteSale	52

8.47	LevelIndicator.....	52
8.48	LocationId.....	53
8.49	LocationIdentifier.....	53
8.50	LockTime.....	53
8.51	MaxAbnormalEvents.....	53
8.52	MostRecentPointer.....	53
8.53	NotOKCounter.....	53
8.54	NumberOfContracts.....	54
8.55	NumberOfEntries.....	54
8.56	NumberOfTimePeriods.....	54
8.57	PermitPeriodOfDay.....	54
8.58	PostCodeld.....	54
8.59	Priority.....	54
8.60	ProductOwner.....	54
8.61	ProductRetailer.....	55
8.62	ProductStatus.....	55
8.63	ReceiptData.....	55
8.64	ReceiptPoint.....	55
8.65	ReservationId.....	55
8.66	RestrictedDayOfWeek.....	55
8.67	RestrictedHalfDayOfWeek.....	56
8.68	RestrictedLocation.....	56
8.69	RestrictedPeriodOfDay.....	56
8.70	RestrictionEnd.....	56
8.71	RestrictionEndDate.....	56
8.72	RestrictionStart.....	56
8.73	SalesPoint.....	56
8.74	SecondaryFlag.....	57
8.75	SectionNumber.....	57
8.76	SecurityVersion.....	57
8.77	SerialNumber.....	57
8.78	ServiceOperator.....	57
8.79	StartDate.....	58
8.80	StartDatePeriod.....	58
8.81	StartDatePeriodStamp.....	58
8.82	StartDateStamp.....	58
8.83	StartTime.....	58
8.84	StartTimeStamp.....	58
8.85	StructureReferenceNumber.....	58
8.86	Surname.....	59
8.87	Telephone.....	59
8.88	TestFlag.....	59
8.89	Time.....	59
8.90	TransactionOperator.....	59
8.91	TransactionSequenceNumber.....	60
8.92	UnblockInstanceNumber.....	60
8.93	UserData.....	60
8.94	ValidationCounter.....	60
8.95	ValidationStatus.....	60
8.96	ValidDayOfExpiry.....	60
8.97	ValidDayOfIssue.....	61
8.98	ValidityCheckFlag.....	61
8.99	ValidityDuration.....	61
8.100	VehicleId.....	61
8.101	VersionNumberFor1545.....	61
9	Encoding rules.....	62
9.1	General.....	62
9.2	Basic encoding rules (BER).....	62
9.3	Alternative encoding rules.....	62

9.3.1	General	62
9.3.2	Packed encoding rules	62
9.3.3	Other encoding rules	62
9.4	Value and size range definitions.....	62
10	Backwards compatibility	63
11	Transport general module definition	63
Annex A (normative)	Assignment of object identifiers	81
Annex B (normative)	Tags.....	82
Annex D (informative)	Index	90

Foreword

This European Standard (EN 1545-1:2005) has been prepared by Technical Committee CEN/TC 224 "Machine readable cards, related device interfaces 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 April 2006, and conflicting national standards shall be withdrawn at the latest by April 2006.

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.

This document supersedes EN 1545-1:1998.

According to the CEN/CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Slovakia, Slovenia, Spain, Sweden, Switzerland and 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 referenced documents are indispensable for the application of this European Standard. 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 3166-1:1997)*

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

ISO 7372, *Trade data interchange -- Trade data elements directory*

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

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

ISO/IEC 7812-1:2000, *Identification cards – Identification of issuers – Part 1: Numbering system*

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: Inter-industry data elements for interchange*

ISO 8824-1/IEC:2002, *Information technology – Abstract Syntax Notation One (ASN.1) : Specification of basic notation*

ISO/IEC 8825-1:2002, *Information technology – ASN.1 encoding rules : Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*

ISO 8825-2/IEC:2002, *Information technology – ASN.1 encoding rules : Specification of Packed Encoding Rules (PER)*

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

3 Terms and definitions

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

3.1

account

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 8824-1:2002

NOTE 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 European Standard, the following abbreviations apply.

AID	Application Identifier [ISO/IEC 7816-5:2004]
ASN.1	Abstract Syntax Notation One [ISO/IEC 8824:2002]
BCD	Binary Coded Decimal
BER	Basic Encoding Rules [ISO/IEC 8825-1:2002]
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 8825-2:2002]
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 : 'yyyyyyymmmmmddddd'B (16 bits)

'yyyyyy'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 explicitly 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 : 'dddddddddddddd'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.

'abcdefghijklmnp'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 '1'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 7812-1:2000 and including the check digit but held as a single integer.

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

Value Assignment : According to ISO/IEC 7812

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 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 $X \times 10^Y$.

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 0162 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 ReferencelIdentifier

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.

ReferencelIdentifier ::= OCTET STRING (SIZE (referencelIdentifierRange))

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 (-8388607.. 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 explicitly 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 : 'mmmmmmmmmmmm'B (11 bits)

6.70 VehicleNumber

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

VehicleNumber ::= LicPlateNumber

6.71 VersionNumber

Identifies the version of a data structure.