

PD CEN ISO/TS 18234-7:2013



BSI Standards Publication

Intelligent transport systems — Traffic and travel information via transport protocol experts group, generation 1 (TPEG1) binary data format

Part 7: Parking information (TPEG1-PKI)

bsi.

...making excellence a habit.
AVA Documents™
Document Sharing Hub

National foreword

This Published Document is the UK implementation of CEN ISO/TS 18234-7:2013.

The UK participation in its preparation was entrusted to Technical Committee EPL/278, Intelligent transport systems.

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.

© The British Standards Institution 2013. Published by BSI Standards Limited 2013

ISBN 978 0 580 80796 1

ICS 03.220.01; 35.240.60

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

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 31 October 2013.

Amendments issued since publication

Date	Text affected
------	---------------

English Version

Intelligent transport systems - Traffic and travel information via transport protocol experts group, generation 1 (TPEG1) binary data format - Part 7: Parking information (TPEG1-PKI) (ISO/TS 18234-7:2013)

Systèmes intelligents de transport - Informations sur le trafic et le tourisme via les données de format binaire du groupe d'experts du protocole de transport, génération 1 (TPEG1) - Partie 7: Informations relatives aux parcs de stationnement (TPEG-PKI) (ISO/TS 18234-7:2013)

Intelligente Transportsysteme - Reise- und Verkehrsinformation über Datenströme der Transportprotokoll Expertengruppe (TPEG) - Teil 7: Parkinformationen (TPEG-PKI) (ISO/TS 18234-7:2013)

This Technical Specification (CEN/TS) was approved by CEN on 15 July 2013 for provisional application.

The period of validity of this CEN/TS is limited initially to three years. After two years the members of CEN will be requested to submit their comments, particularly on the question whether the CEN/TS can be converted into a European Standard.

CEN members are required to announce the existence of this CEN/TS in the same way as for an EN and to make the CEN/TS available promptly at national level in an appropriate form. It is permissible to keep conflicting national standards in force (in parallel to the CEN/TS) until the final decision about the possible conversion of the CEN/TS into an EN is reached.

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

Foreword

This document (CEN ISO/TS 18234-7:2013) has been prepared by Technical Committee ISO/TC 204 "Intelligent transport systems" in collaboration with Technical Committee CEN/TC 278 "Intelligent transport systems" the secretariat of which is held by NEN.

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.

According to the CEN-CENELEC Internal Regulations, the national standards organizations of the following countries are bound to announce this Technical Specification: 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.

Endorsement notice

The text of ISO/TS 18234-7:2013 has been approved by CEN as CEN ISO/TS 18234-7:2013 without any modification.

Contents

Page

Foreword	v
Introduction.....	vii
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Abbreviated terms	2
5 Application identification.....	2
6 Service Component Frame	3
7 Message Components	3
7.1 List of Generic Component Ids	3
7.2 Parking Message	4
7.2.1 MessageManagement	5
7.2.2 ParkingLocation	6
7.2.3 ParkingSiteDescription	6
7.2.4 CurrentCapacity.....	23
7.2.5 ExpectedCapacity.....	25
7.2.6 Advice	26
8 Parking Information Tables	27
8.1 Structure and semantics.....	27
8.2 Indexing	27
8.3 CEN-English 'Word', Comments and Examples	28
8.3.1 pki001:VehicleType	28
8.3.2 pki002:ParkingType.....	29
8.3.3 pki003:UserType.....	30
8.3.4 pki004:FuelType	31
8.3.5 pki005:AvailableFeatures	31
8.3.6 pki006:EventType	32
8.3.7 pki007:Reservability	32
8.3.8 pki008:FacilityType	33
8.3.9 pki009:SupervisionType	33
8.3.10 pki010:SecurityType.....	34
8.3.11 pki011:AssociatedService	34
8.3.12 pki012:ParkingStatus	35
8.3.13 pki013:PaymentMethod	35
8.3.14 pki014:SiteServed.....	36
8.3.15 pki015:GateType.....	36
8.3.16 pki016:ContactType	37
8.3.17 pki017:TransportType	37
8.3.18 pki018:OpeningHoursType.....	38
8.3.19 pki019:TermType	38
8.3.20 pki020:Advice	39
8.3.21 pki021:Tendency	39
8.3.22 pki022:FeeType.....	40
Annex A (normative) Binary SSF and Data Types.....	41
A.1 Conventions and symbols.....	41
A.1.1 Conventions	41
A.1.2 Symbols.....	41

A.2	Representation of syntax	42
A.2.1	General.....	42
A.2.2	Data type notation.....	42
A.2.3	Application dependent data types	45
A.2.4	Toolkits and external definition.....	50
A.2.5	Application design principles.....	50
A.3	TPEG data stream description	51
A.3.1	Diagrammatic hierarchy representation of frame structure.....	51
A.3.2	Syntactical Representation of the TPEG Stream.....	51
A.3.3	Description of data on Transport level.....	56
A.3.4	Description of data on Service level.....	57
A.3.5	Description of data on Service component level	58
A.4	General binary data types	58
A.4.1	Primitive data types	58
A.4.2	Compound data types	64
A.4.3	Table definitions	66
A.4.4	Tables.....	68
Annex B	(normative) TPEG Message Management Container, MMC (Binary).....	84
B.1	Terms and Definitions	84
B.1.1	Message.....	84
B.1.2	Monolithic Message Management.....	84
B.1.3	Multi-Part Message Management.....	84
B.1.4	Top Level Container	84
B.2	Symbols (and abbreviated terms).....	84
B.2.1	MMC	84
B.2.2	PKI	84
B.3	Introduction	84
B.4	Message Components.....	85
B.4.1	MMCTemplate.....	85
B.4.2	MessageManagementContainer.....	87
B.4.3	MMCMasterMessage	88
B.4.4	MMCMessagePart	90
B.5	Datatypes.....	91
B.5.1	MultiPartMessageDirectory	91
B.6	Tables.....	91
B.6.1	Structure and semantics	91
B.6.2	Indexing	92
B.6.3	Codes, Names and Comments	92
Bibliography		94

Foreword

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

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

In other circumstances, particularly when there is an urgent market requirement for such documents, a technical committee may decide to publish other types of normative document:

- an ISO Publicly Available Specification (ISO/PAS) represents an agreement between technical experts in an ISO working group and is accepted for publication if it is approved by more than 50 % of the members of the parent committee casting a vote;
- an ISO Technical Specification (ISO/TS) represents an agreement between the members of a technical committee and is accepted for publication if it is approved by 2/3 of the members of the committee casting a vote.

An ISO/PAS or ISO/TS is reviewed after three years in order to decide whether it will be confirmed for a further three years, revised to become an International Standard, or withdrawn. If the ISO/PAS or ISO/TS is confirmed, it is reviewed again after a further three years, at which time it must either be transformed into an International Standard or be withdrawn.

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

ISO/TS 18234-7 was prepared by the European Committee for Standardization (CEN) Technical Committee CEN/TC 278, *Road transport and traffic telematics*, in collaboration with ISO Technical Committee ISO/TC 204, *Intelligent transport systems*, in accordance with the Agreement on technical cooperation between ISO and CEN (Vienna Agreement).

ISO/TS 18234 consists of the following parts, under the general title *Intelligent transport systems — Traffic and travel information via transport protocol experts group, generation 1 (TPEG1) binary data format*:

- *Part 1: Introduction, numbering and versions (TPEG1-INV)*
- *Part 2: Syntax, semantics and framing structure (TPEG1-SSF)*
- *Part 3: Service and network information (TPEG1-SNI)*
- *Part 4: Road Traffic Message application (TPEG1-RTM)*
- *Part 5: Public Transport Information (PTI) application*
- *Part 6: Location referencing applications*

- *Part 7: Parking information (TPEG1-PK1)*
- *Part 8: Congestion and travel-time application (TPEG1-CTT)*
- *Part 9: Traffic event compact (TPEG1-TEC)*
- *Part 10: Conditional access information (TPEG1-CAI)*
- *Part 11: Location Referencing Container (TPEG1-LRC)*

Introduction

TPEG technology uses a byte-oriented data stream format, which may be carried on almost any digital bearer with an appropriate adaptation layer. TPEG messages are delivered from service providers to end-users, and are used to transfer information from the database of a service provider to a user's equipment.

The brief history of TPEG technology development dates back to the European Broadcasting Union (EBU) Broadcast Management Committee establishing the B/TPEG project group in autumn 1997 with the mandate to develop, as soon as possible, a new protocol for broadcasting traffic and travel-related information in the multimedia environment. TPEG technology, its applications and service features are designed to enable travel-related messages to be coded, decoded, filtered and understood by humans (visually and/or audibly in the user's language) and by agent systems.

One year later in December 1998, the B/TPEG group produced its first EBU specifications. Two Technical Specifications were released. ISO/TS 18234-2 (TPEG-SSF) described the Syntax, Semantics and Framing Structure, which is used for all TPEG applications. ISO/TS 18234-4 (TPEG-RTM) described the first application, for Road Traffic Messages.

Subsequently, CEN/TC 278/WG 4, in conjunction with ISO/TC 204, established a project group comprising the members of B/TPEG and they have continued the work concurrently since March 1999. Since then two further parts were developed to make the initial complete set of four parts, enabling the implementation of a consistent service. ISO/TS 18234-3 describes the Service and Network Information Application, which should be used by all service implementations to ensure appropriate referencing from one service source to another. ISO/TS 18234-1 completes the series, by describing the other parts and their relationship; it also contains the application IDs used within the other parts. Additionally, ISO/TS 18234-5, the Public Transport Information Application (TPEG-PTI) and ISO/TS 18234-6 (TPEG-LRC), were developed.

This Technical Specification adds another powerful application to the ISO 18234-series allowing detailed parking information to be encoded and transmitted to the user. This Technical Specification includes new advanced message management and new datatypes, as specified in the annexes.

Today, traffic congestion has become a serious problem in urban areas. Some traffic congestion is attributed to drivers searching for parking spaces. Therefore, timely provision of parking information could help ease traffic congestion. Furthermore, parking information would be valuable for the visitor, particularly when it could be used to signal where a temporary parking facility is established for a special occasion.

TPEG applications are developed using UML modelling and a software tool is used to automatically select content which then populates this Technical Specification. Diagrammatic extracts from the model are used to show the capability of the binary coding in place of lengthy text descriptions; the diagrams do not necessarily include all relevant content possible.

This Technical Specification describes the binary data format of the on-air interface of the Parking Information application, (TPEG-PKI) with the technical version number TPEG-PKI_1.0/001.

Intelligent transport systems — Traffic and travel information via transport protocol experts group, generation 1 (TPEG1) binary data format —

Part 7: Parking information (TPEG1-PKI)

1 Scope

This Technical Specification specifies the TPEG Parking Information Application (PKI) which is designed to deliver parking information to a variety of receivers using a number of different channels, foremost digital broadcasting and internet technologies. Parking information may be presented to the user in many different ways including textually, voiced and graphically using standard formats.

2 Normative references

The following referenced documents are indispensable for the application of this Technical Specification. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 639-1:2002, *Codes for the representation of names of languages — Part 1: Alpha-2 code*

ISO 3166-1:2006, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*

ISO 4217:2001, *Codes for the representation of currencies and funds*

ISO/TS 18234-1, *Intelligent transport systems — Traffic and travel information via transport protocol experts group, generation 1 (TPEG1) binary data format — Part 1: Introduction, numbering and versions (TPEG1-INV)*

ISO/TS 18234-2, *Intelligent transport systems — Traffic and travel information via transport protocol experts group, generation 1 (TPEG1) binary data format — Part 2: Syntax, semantics and framing structure (SSF)*

ISO/TS 18234-3, *Intelligent transport systems — Traffic and travel information via transport protocol experts group, generation 1 (TPEG1) binary data format — Part 3: Service and network information (TPEG1-SNI)*

ISO/TS 18234-11, *Intelligent transport systems — Traffic and travel information via transport protocol experts group, generation 1 (TPEG1) binary data format — Part 11: Location Referencing Container (TPEG1-LRC)*

IEC 60559:1989, *Binary floating-point arithmetic for microprocessor systems*

3 Terms and definitions

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

3.1 application identification AID

application that indicates how to process TPEG content and routes information to the appropriate application decoder

NOTE Each TPEG application has a unique number, which identifies the application according to Clause 5 of this Technical Specification. The application identification is part of the TPEG specification and is defined as and when new applications are developed.

4 Abbreviated terms

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

AID	Application Identification
B/TPEG	Broadcast/TPEG (the EBU project group name for the specification drafting group)
CEN	Comité Européen de Normalisation
EBU	European Broadcasting Union
IPR	Intellectual Property Right(s)
MMC	Message Management Container Toolkit
PKI	Parking Information Application
PTI	Public Transport Information
RTM	Road Traffic Message
TEC	Traffic Event Compact application
TPEG	Transport Protocol Expert Group
TTI	Traffic and Travel Information

5 Application identification

The word 'application' is used in the TPEG specifications to describe specific subsets of the TPEG structure. An application defines a limited vocabulary for a certain type of messages, for example parking information or road traffic information. Each TPEG application is assigned a unique number, called the Application Identification (AID). An AID is defined whenever a new application is developed and these are all listed in CEN ISO/TS 18234-1.

The application identification number is used within the TPEG-SNI application to indicate how to process TPEG content and facilitates the routing of information to the appropriate application decoder.

TPEG-PKI is assigned the AID = 0003 hex.

6 Service Component Frame

PKI makes use of the "Service component frame with dataCRC, groupPriority, and messageCount" according to Annex A, Clause A.3.2.6.2.4. For explanatory purpose this is repeated here.

< ServCompFramePrioritisedCountedProtected>:=	: CRC protected service component frame with group priority and message count
<ServCompFrameHeader> (header),	: Component frame header as defined in A.3.2.6.
<typ007:Priority> (groupPriority),	: group priority applicable to all messages in the ApplicationContent
<IntUnTi> (messageCount),	: count of messages in this ApplicationContent
external <ApplicationContent> (content),	: actual payload of the application
<CRC> (dataCRC);	: CRC starting with first byte after the header

The main frame of PKI defines ApplicationContent as follows:

<ApplicationContent>:=	: Service component frame template
messageCount * <ParkingMessage> (msg);	: derived header from [SSF], AID = 3
	: Any number of any PKI message components

7 Message Components

7.1 List of Generic Component Ids

Name	Id
Parking Message	0
MessageManagementContainer	1
MMCMasterMessage	2
MMCMessagePart	3
ParkingLocation	4
ParkingSiteDescription	5
CurrentCapacity	6
CurrentCapacityFor	7
ExpectedCapacity	8
ExpectedCapacityFor	9
InformationFor	10
SizeRestrictions	11
ParkingInfo	12
ParkingSpecification	13
Logo	14
Contact	16
OpeningHours	17
GateInfo	18
PricingPayment	19
PaymentDetails	20

Facilities	21
ToSite	23
Advice	24
AssociatedService	25
ParkingForEvent	26

7.2 Parking Message

A parking message shall contain a MessageManagement component and should have ParkingLocation, ParkingSiteDescription and Advice components as well as one CurrentCapacity and several ExpectedCapacity components, as shown in Figure 1.

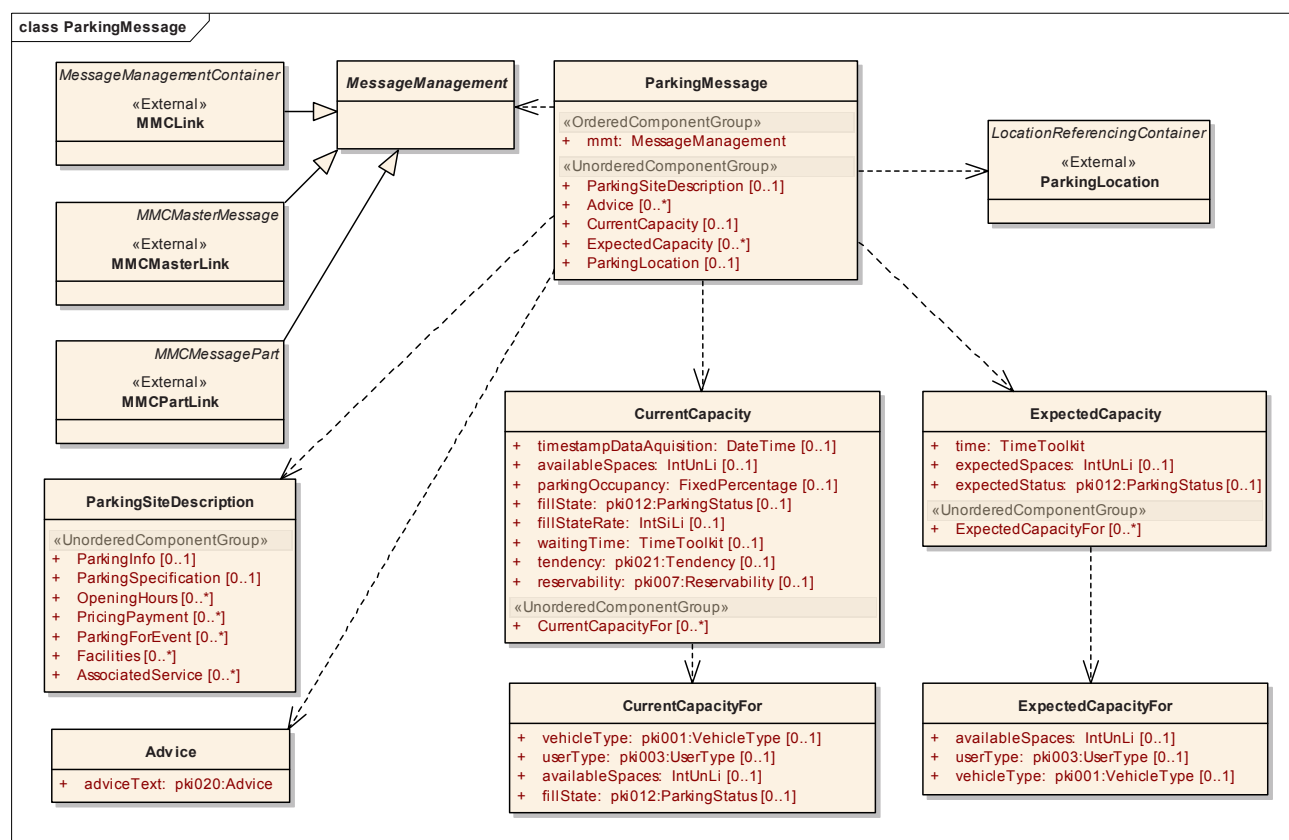


Figure 1 — Structure of a parking message component

Methods of the TPEG Message Management Container Annex B - Message Management Container can be used to transmit static data independent from the dynamic data. The components have been grouped to easily allow such dynamic updates.

For example the name and the location of a parking site do not change frequently and thus this information can be transmitted less frequently, unlike the number of available spaces. It is important none the less that the basic information required to display a sensible message to the user should be sent in suitable intervals to allow receivers just switched on to decode data within reasonable time.

Clients should decode messages with the same version number (and PartID in case of partial messages) only once.

```

<ParkingMessage<Component(0)>>:=

    <IntUnTi>(0),                                : Identifier = 0

    <IntUnLoMB>(lengthComp),                      : Length of component in bytes, excluding the id and length
                                                    indicator

    <IntUnLoMB>(lengthAttr),                      : Length of attributes, always 0 since this component has no
                                                    attributes

    <MessageManagement>(mmt),                   : Message Management Container

unordered {

    m * <ParkingLocation> [0..1],

    m * <ParkingSiteDescription> [0..1],

    m * <CurrentCapacity> [0..1],

    m * <ExpectedCapacity> [0..*],

    m * <Advice> [0..*]

};

```

7.2.1 MessageManagement

Serves as an optional link to the standard MessageManagementContainer. See Annex B - Message Management Container for content.

```

<MessageManagement(x)<Component(x)>>:=

    <IntUnTi>(x),                                : Identifier, is defined by instance

    <IntUnLoMB>(lengthComp),                      : Length of component in bytes, excluding the id and length
                                                    indicator

    <IntUnLoMB>(lengthAttr);                      : Length of attributes of this component in bytes

```

The purpose of the following definitions is to assign a unique identifier to the components.

<MMCLink<MessageManagement(1)>>:=

external **<MessageManagementContainer(1)>;** : Identifier = 1

<MMCMasterLink<MessageManagement(2)>>:=

external **<MMCMasterMessage(2)>;** : Identifier = 2

<MMCPartLink<MessageManagement(3)>>:=

external **<MMCMessagesPart(3)>;** : Identifier = 3

7.2.2 ParkingLocation

Serves as a link to the LocationReferencingContainer, described in ISO/TS 18234-11

The purpose of this component definition is to assign a unique identifier to the component.

<ParkingLocation<LocationReferencingContainer(4)>>:=

external **<LocationReferencingContainer(4)>;** : Identifier = 4

7.2.3 ParkingSiteDescription

The ParkingSiteDescription component is a wrapper for mostly static information about a parking facility. The information is grouped in the ParkingInfo, ParkingSpecification, OpeningHours, PricingPayment, Facilities, ParkingForEvent and AssociatedService components, as shown in Figure 2.

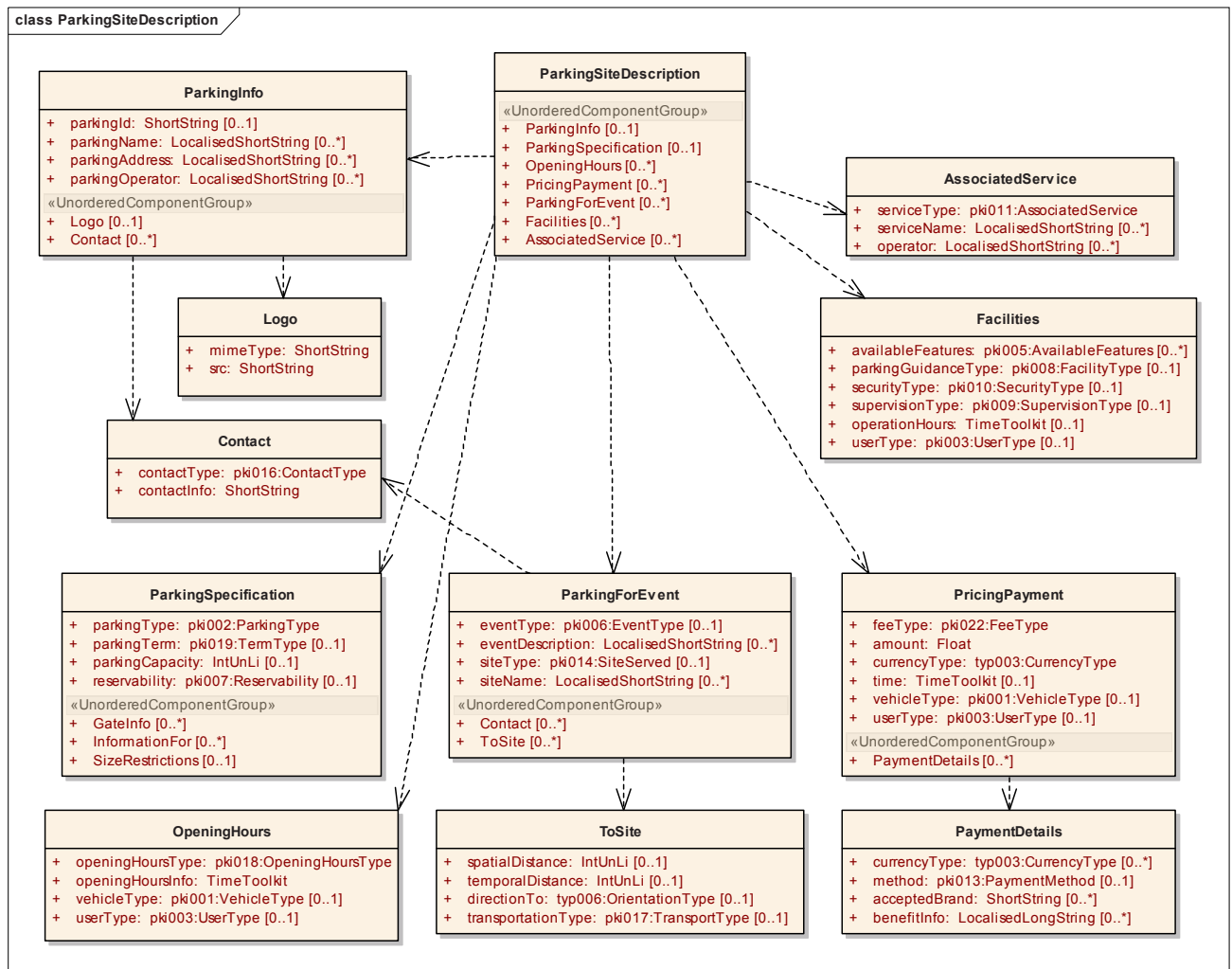


Figure 2 — Structure of the ParkingSiteDescription

<ParkingSiteDescription<Component(5)>>:=

<IntUnTi>(5), : Identifier = 5

<IntUnLoMB>(lengthComp), : Length of component in bytes, excluding the id and length indicator

<IntUnLoMB>(lengthAttr), : Length of attributes, always 0 since this component has no attributes

unordered {

m * **<ParkingInfo>** [0..1],

m * **<ParkingSpecification>** [0..1],

m * **<ParkingForEvent>** [0..*],

m * **<OpeningHours>** [0..*],

m * **<PricingPayment>** [0..*],

m * **<Facilities>** [0..*],

m * **<AssociatedService>** [0..*]

};

7.2.3.1 ParkingInfo

The ParkingInfo component contains address and contact information about a parking facility to be displayed to the user. This includes name, address, operator, logo and the contact details.

```

<ParkingInfo<Component(12)>>:=
    <IntUnTi>(12),                : Identifier = 12
    <IntUnLoMB>(lengthComp),       : Length of component in bytes, excluding the id and length
                                   : indicator
    <IntUnLoMB>(lengthAttr),       : Length of attributes of this component in bytes
    <BitArray>(selector),         : 1 byte containing 4 switches.

    If (bit 0 of selector is set)
        <ShortString>(parkingId), : This attribute may hold a parking site specific ID string,
                                   : allowing linking this site to other referencing schemes. It
                                   : shall not contain language specific descriptions and should
                                   : preferably not be presented to the user as a description.

    If (bit 1 of selector is set) {
        <IntUnLoMB>(n),            : Number of entries in array attribute, between 0 and infinity.
        n * <LocalisedShortString>(parkingName) [0..*] : Name of the parking in various languages.
    }

    If (bit 2 of selector is set) {
        <IntUnLoMB>(n),            : Number of entries in array attribute, between 0 and infinity.
        n * <LocalisedShortString>(parkingAddress) [0..*] : Address of the parking in language specific formats.
                                                         : e.g. for German: Frauensteige 2, D-89075 Ulm
    }

    If (bit 3 of selector is set) {
        <IntUnLoMB>(n),            : Number of entries in array attribute, between 0 and infinity.
        n * <LocalisedShortString>(parkingOperator) [0..*], : Language specific strings representing the name and/or
                                                         : company of the operator.
    }

    unordered {
        m * <Logo> [0..1],         : m represents the number of occurrences, between zero
                                   : and one.
        m * <Contact> [0..*]       : m represents the number of occurrences, between 0 and
                                   : infinity.
    };

```

7.2.3.1.1 Logo

The logo component provides a URL and a mimeType for a company or parking site logo type. It does not contain the image data itself.

<Logo<Component(14)>>:=

<IntUnTi>(14),	: Identifier = 14
<IntUnLoMB>(lengthComp),	: Length of component in bytes, excluding the id and length indicator
<IntUnLoMB>(lengthAttr),	: Length of attributes of this component in bytes
<ShortString>(mimeType),	: The mime type of the image at the provided source.
<ShortString>(src);	: URL where the logo can be retrieved.

7.2.3.1.2 Contact

Provides contact information.

<Contact<Component(16)>>:=

<IntUnTi>(16),	: Identifier = 16
<IntUnLoMB>(lengthComp),	: Length of component in bytes, excluding the id and length indicator
<IntUnLoMB>(lengthAttr),	: Length of attributes of this component in bytes
<pki016:ContactType>(contactType),	: Indicates the type of the information in the contactInfo attribute.
<ShortString>(contactInfo);	: The actual contact information of the type indicated in the contactType attribute.

7.2.3.2 ParkingSpecification

The ParkingSpecification component contains detailed and mostly static information about a parking facility describing properties of the parking site. This includes the type of parking and the maximum capacity among other information, as shown in Figure 3.

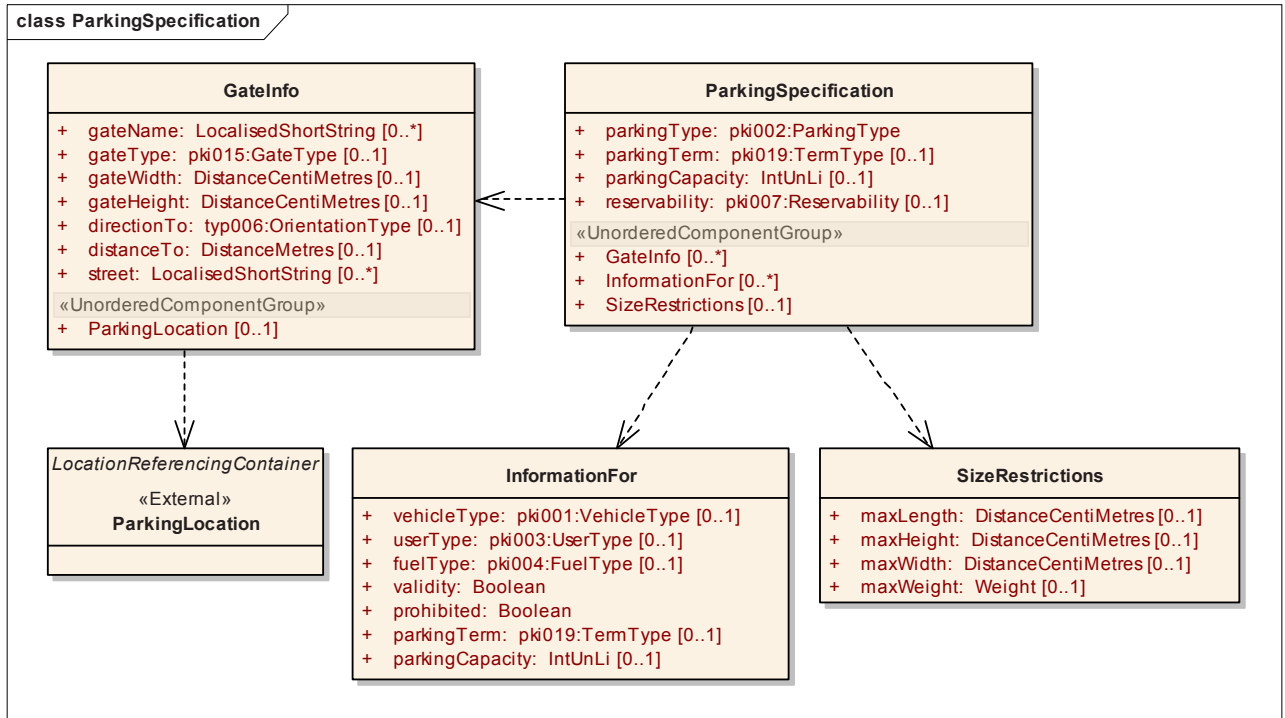


Figure 3 — Structure of the ParkingSpecification

<ParkingSpecification<Component(13)>>:=

<IntUnTi>(13),	: Identifier = 13
<IntUnLoMB>(lengthComp),	: Length of component in bytes, excluding the id and length indicator
<IntUnLoMB>(lengthAttr),	: Length of attributes of this component in bytes
<pk002:ParkingType>(parkingType),	: Indicates the over all type for the parking.
<BitArray>(selector),	: 1 byte containing 3 switches.

If (bit 0 of selector is set)

<pk019:TermType>(parkingTerm),	: Classifies the site with respect to the maximum allowed parking time or pricing schema. For example short term parking sites are usually very expensive for long term use. For the rare case that a parking site offers places for several term types, the parking site should be represented as one separate "virtual" parking site for each term type.
---	--

If (bit 1 of selector is set)

<IntUnLi>(parkingCapacity),	: Total number of max available spaces.
--	---

If (bit 2 of selector is set)

<pk007:Reservability>(reservability),	: Indicates whether it is possible to reserve places.
--	---

unordered {

m * <InformationFor> [0..*],	: m represents the number of occurrences, between 0 and infinity.
m * <SizeRestrictions> [0..1],	: m represents the number of occurrences, between zero and one.
m * <GateInfo> [0..*]	: m represents the number of occurrences, between 0 and infinity.

};

7.2.3.2.1 InformationFor

The InformationFor component contains more detailed information about specific vehicle types, user groups or fuel types. This component allows targeted information to be delivered to particular groups, such as disabled drivers for example, with regard to the parking term, the number of reserved spaces, the maximum number of available spaces and other specific information. The component can further signal who may or may not use the particular site. The number of reserved spaces can e.g. be encoded as a component with userType "reservation holders".

The numbers contained within this component are a subset of the numbers indicated in the attributes of the ParkingSpecification component. The sum of the numbers might be larger than the total number of available spaces, because there may be overlapping groups.

For example all of the spaces available for a certain user group, might be also available for a special vehicle type.

<InformationFor<Component(10)>>:=

<IntUnTi>(10),	: Identifier = 10
<IntUnLoMB>(lengthComp),	: Length of component in bytes, excluding the id and length indicator
<IntUnLoMB>(lengthAttr),	: Length of attributes of this component in bytes
<BitArray>(selector),	: 1 byte containing 7 switches.

If (bit 0 of selector is set)

<pki001:VehicleType>(vehicleType),	: This attribute indicates the vehicle type the information in the component is valid for.
---	--

If (bit 1 of selector is set)

<pki003:UserType>(userType),	: This attribute indicates the user type the information in the component is valid for.
---	---

If (bit 2 of selector is set)

<pki004:FuelType>(fuelType),	: This attribute indicates the fuel type a vehicle must use for the information in this component to be valid.
---	--

If (bit 3 of selector is set)

<Boolean>(validity),	: If set to true, bit 4 contains valid information. If set to false, bit 4 shall be ignored by the decoder.
-----------------------------------	---

If (bit 4 of selector is set)

<Boolean>(prohibited),	: If set to true, the usage of this site is prohibited for groups indicated by the above attributes. If set to false the use is explicitly allowed.
-------------------------------------	---

If (bit 5 of selector is set)

<pki019:TermType>(parkingTerm),	: Holds information about the term the parking is available for the indicated group.
--	--

If (bit 6 of selector is set)

<IntUnLi>(parkingCapacity);	: Number of spaces available for the indicated group.
--	---

7.2.3.2.2 SizeRestrictions

The SizeRestrictions component defines physical limits concerning the maximum size of vehicles that can use the parking site.

<SizeRestrictions<Component(11)>>:=

<IntUnTi>(11),	: Identifier = 11
<IntUnLoMB>(lengthComp),	: Length of component in bytes, excluding the id and length indicator
<IntUnLoMB>(lengthAttr),	: Length of attributes of this component in bytes
<BitArray>(selector),	: 1 byte containing 4 switches.
If (bit 0 of selector is set)	
<DistanceCentiMetres>(maxLength),	: The maximum length in centimetres.
If (bit 1 of selector is set)	
<DistanceCentiMetres>(maxHeight),	: The maximum height in centimetres.
If (bit 2 of selector is set)	
<DistanceCentiMetres>(maxWidth),	: The maximum width in centimetres.
If (bit 3 of selector is set)	
<Weight>(maxWeight);	: The maximum weight in kilogrammes.

7.2.3.2.3 GateInfo

GateInfo provides further details for individual gates. It can hold a LocationReferenceContainer to accurately identify the location of the specific gate.

```

<GateInfo<Component(18)>>:=
    <IntUnTi>(18),                                : Identifier = 18
    <IntUnLoMB>(lengthComp),                       : Length of component in bytes, excluding the id and length
                                                    : indicator
    <IntUnLoMB>(lengthAttr),                       : Length of attributes of this component in bytes
    <BitArray>(selector),                         : 1 byte containing 7 switches.
    If (bit 0 of selector is set) {
        <IntUnLoMB>(n),
        n * <LocalisedShortString>(gateName) [0..*] : Language specific names for the gate to be presented to
                                                    : the user.
    }
    If (bit 1 of selector is set)
        <pki015:GateType>(gateType),               : Specifies the type of the gate.
    If (bit 2 of selector is set)
        <DistanceCentiMetres>(gateWidth),          : The minimum width of the gate.
    If (bit 3 of selector is set)
        <DistanceCentiMetres>(gateHeight),         : The minimum height of the gate.
    If (bit 4 of selector is set)
        <typ006:OrientationType>(directionTo),     : Direction to the street listed in the street attribute. For
                                                    : example the next street from the gate is in direction "north".
    If (bit 5 of selector is set)
        <DistanceMetres>(distanceTo),              : The distance to the street listed in the street attribute. For
                                                    : example the next street from the gate is 1500 m away in the
                                                    : above mentioned direction.
    If (bit 6 of selector is set) {
        <IntUnLoMB>(n),
        n * <LocalisedShortString>(street) [0..*] : The language specific names of a street next to the gate.
                                                    : Only one street must be named, although it can be named
                                                    : in various languages.
    }
    unordered {
        m * <ParkingLocation> [0..1]
    };

```

7.2.3.2.3.1 ParkingLocation

See above Clause 7.2.2.

7.2.3.3 ParkingForEvent

Parking sites sometimes offer places for visitors of major events or famous places, or are at times even especially set up for a specific event. The type and name of the event can be indicated within this component. Additional details about the event would be expected to be delivered via other TPEG applications. A separate component has to be used for each event.

<ParkingForEvent<Component(26)>>:=

<IntUnTi>(26),	: Identifier = 26
<IntUnLoMB>(lengthComp),	: Length of component in bytes, excluding the id and length indicator
<IntUnLoMB>(lengthAttr),	: Length of attributes of this component in bytes
<BitArray>(selector),	: 1 byte containing 4 switches.

If (bit 0 of selector is set)

<pki006:EventType>(eventType),	: Allows indication of what kind of event the parking is for.
---	---

If (bit 1 of selector is set) {

<IntUnLoMB>(n),	
n * <LocalisedShortString>(eventDescription) [0..*]	: A language specific name or description can be included to further specify the related event.

}

If (bit 2 of selector is set)

<pki014:SiteServed>(siteType),	: The type of the site for which this parking is for can be given.
---	--

If (bit 3 of selector is set) {

<IntUnLoMB>(n),	
n * <LocalisedShortString>(siteName) [0..*]	: A language specific name or description can be included to further specify the related site.

}

unordered {

m * <Contact> [0..*],	
m * <ToSite> [0..*]	

};

7.2.3.3.1 Contact

See above 7.1.3.1.2.

7.2.3.3.2 ToSite

Information on how the related site can be reached.

<ToSite<Component(23)>>:=

<IntUnTi>(23),	: Identifier = 23
<IntUnLoMB>(lengthComp),	: Length of component in bytes, excluding the id and length indicator
<IntUnLoMB>(lengthAttr),	: Length of attributes of this component in bytes
<BitArray>(selector),	: 1 byte containing 4 switches.

If (bit 0 of selector is set)

<IntUnLi>(spatialDistance),	: The distance in metres to reach the related facility.
--	---

If (bit 1 of selector is set)

<IntUnLi>(temporalDistance),	: Time in minutes it takes to the related facility using the indicated transportation.
---	--

If (bit 2 of selector is set)

<typ006:OrientationType>(directionTo),	: The direction to the related facility.
---	--

If (bit 3 of selector is set)

<pk017:TransportType>(transportationType);	: The means of transportation for which the information in this component is valid.
---	---

7.2.3.4 OpeningHours

Times the parking is open with respect to vehicle and user type.

<OpeningHours<Component(17)>>:=

<IntUnTi>(17),	: Identifier = 17
<IntUnLoMB>(lengthComp),	: Length of component in bytes, excluding the id and length indicator
<IntUnLoMB>(lengthAttr),	: Length of attributes of this component in bytes
<pki018:OpeningHoursType>(openingHoursType),	: Defines how the openingHours attribute has to be interpreted.
<TimeToolkit>(openingHoursInfo),	: The actual timespan encoded using the TimeToolkit container.
<BitArray>(selector),	: 1 byte containing 2 switches.

If (bit 0 of selector is set)

<pki001:VehicleType>(vehicleType),	: The vehicle type for which the information within this component is valid.
------------------------------------	--

If (bit 1 of selector is set)

<pki003:UserType>(userType);	: The information in this component is relevant for these users.
------------------------------	--

7.2.3.5 PricingPayment

The PricingPayment component describes the costs for parking at this parking location as well as further payment details.

<PricingPayment<Component(19)>>:=	
<IntUnTi>(19),	: Identifier = 19
<IntUnLoMB>(lengthComp),	: Length of component in bytes, excluding the id and length indicator
<IntUnLoMB>(lengthAttr),	: Length of attributes of this component in bytes
<pki022:FeeType>(feeType),	: Defines what sort of fee is described within this component.
<Float>(amount),	: The actual amount of the fee, expressed in the currency specified in the currencyType attribute.
<typ003:CurrencyType>(currencyType),	: The currency in which the amount of the fee is given.
<BitArray>(selector),	: 1 byte containing 3 switches.
If (bit 0 of selector is set)	
<TimeToolkit>(time),	: Time period for which this pricing information applies.
If (bit 1 of selector is set)	
<pki001:VehicleType>(vehicleType),	: Limits the information to a vehicle type.
If (bit 2 of selector is set)	
<pki003:UserType>(userType),	: Limits the information to a user type.
unordered {	
m * <PaymentDetails> [0..*]	
};	

7.2.3.5.1 PaymentDetails

Additional details on currencies, methods of payment and benefit information.

```

<PaymentDetails<Component(20)>>:=

    <IntUnTi>(20),                                : Identifier = 20

    <IntUnLoMB>(lengthComp),                        : Length of component in bytes, excluding the id and length
                                                    : indicator

    <IntUnLoMB>(lengthAttr),                        : Length of attributes of this component in bytes

    <BitArray>(selector),                          : 1 byte containing 4 switches.

    If (bit 0 of selector is set) {

        <IntUnLoMB>(n),                            : Number of entries in array attribute, between 0 and infinity.

        n * <typ003:CurrencyType>(currencyType) [0..*], : Payment is accepted in these currencies.

    }

    If (bit 1 of selector is set)

        <pki013:PaymentMethod>(method),            : How the fee can be paid.

    If (bit 2 of selector is set) {

        <IntUnLoMB>(n),                            : Number of entries in array attribute, between 0 and infinity.

        n * <ShortString>(acceptedBrand) [0..*],    : Brands, products and company names that are accepted
                                                    : for the specified method of payment.

    }

    If (bit 3 of selector is set) {

        <IntUnLoMB>(n),                            : Number of entries in array attribute, between 0 and infinity.

        n * <LocalisedLongString>(benefitInfo) [0..*] : Language specific announcement of a special benefit.

    };

```

7.2.3.6 Facilities

Indicates facilities and features, such as guided parking, mechanical parking which are available at the parking site and their operating hours as well as the user type.

Several facilities can be encoded in the attributes of one Facilities component at the same time, if they for example have identical operation hours.

<Facilities<Component(21)>>:=	
<IntUnTi>(21),	: Identifier = 21
<IntUnLoMB>(lengthComp),	: Length of component in bytes, excluding the id and length indicator
<IntUnLoMB>(lengthAttr),	: Length of attributes of this component in bytes
<BitArray>(selector),	: 1 byte containing 6 switches.
If (bit 0 of selector is set) {	
<IntUnLoMB>(n),	: Number of entries in array attribute, between 0 and infinity.
n * <pki005:AvailableFeatures> \	: Features that are present at the parking facility can be listed here.
\ (availableFeatures) [0..*],	
}	
If (bit 1 of selector is set)	
<pki008:FacilityType>(parkingGuidanceType),	: The information is related to the parking guidance specified here.
If (bit 2 of selector is set)	
<pki010:SecurityType>(securityType),	: The information concerns this security type.
If (bit 3 of selector is set)	
<pki009:SupervisionType>(supervisionType),	: The information is related to this supervision type.
If (bit 4 of selector is set)	
<TimeToolkit>(operationHours),	: This attribute is used to indicate the operation time for the information provided in the other attributes of this component.
If (bit 5 of selector is set)	
<pki003:UserType>(userType);	: The facility is relevant for this user type.

7.2.3.7 AssociatedService

Carries the description of associated services which are available at the parking site and usually have a name. Simple services that usually do not have a name may be indicated in the availableFeatures attribute in the Facilities component.

<AssociatedService<Component(25)>>:=

<IntUnTi>(25),	: Identifier = 25
<IntUnLoMB>(lengthComp),	: Length of component in bytes, excluding the id and length indicator
<IntUnLoMB>(lengthAttr),	: Length of attributes of this component in bytes
<pk011:AssociatedService>(serviceType),	: The type of the associated service.
<BitArray>(selector),	: 1 byte containing 2 switches.

If (bit 0 of selector is set) {

<IntUnLoMB>(n),	: Number of entries in array attribute, between 0 and infinity.
n * <LocalisedShortString>(serviceName) [0..*],	: Description of the service in relevant languages.

}

If (bit 1 of selector is set) {

<IntUnLoMB>(n),	: Number of entries in array attribute, between 0 and infinity.
n * <LocalisedShortString>(operator) [0..*]	: Language specific strings representing the name and/or company of the operator.

};

7.2.4 CurrentCapacity

The CurrentCapacity element describes the current parking situation by means of the exact number of available spaces, tendencies and qualitative descriptions.

This component indicates the overall number of parking spaces available at the time of message generation. Contained components may specify the number and type of spaces the capacity refers to.

<CurrentCapacity<Component(6)>>:=	
<IntUnTi>(6),	: Identifier = 6
<IntUnLoMB>(lengthComp),	: Length of component in bytes, excluding the id and length indicator
<IntUnLoMB>(lengthAttr),	: Length of attributes of this component in bytes
<BitArray>(selector),	: 2 bytes containing 8 switches.
If (bit 0 of selector is set)	
<DateTime>(timestampDataAquisition),	: The date and time when data has been measured or acquired.
If (bit 1 of selector is set)	
<IntUnLi>(availableSpaces),	: The total number of available spaces. This is a quantitative measure
If (bit 2 of selector is set)	
<FixedPercentage>(parkingOccupancy),	: Number of occupied spaces in relation to the capacity specified in ParkingSpecification (0-100%).
If (bit 3 of selector is set)	
<pki012:ParkingStatus>(fillState),	: Describes the overall status of the site with respect to parking occupancy.
If (bit 4 of selector is set)	
<IntSiLi>(fillStateRate),	: The rate (cars per hour) at which the site is filling. Negative numbers indicate that cars are leaving the site.
If (bit 5 of selector is set)	
<TimeToolkit>(waitingTime),	: The time it takes to actually enter the parking.
If (bit 6 of selector is set)	
<pki021:Tendency>(tendency),	: The currently observed tendency. This is no prognosis.
If (bit 7 of selector is set)	
<pki007:Reservability>(reservability),	
unordered {	
m * <CurrentCapacityFor> [0..*];	
}	