
**Road Transport and Traffic Telematics
(RTTT) — Electronic Fee Collection (EFC) —
Application interface definition for
dedicated short range communications**

*Télématique de la circulation et du transport routier — Perception du
télépéage — L'interface applicative relative aux communications dédiées
aux courtes portées*



Foreword

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

The main task of technical committees is to prepare International Standards, but in exceptional circumstances a technical committee may propose the publication of a Technical Report of one of the following types:

- type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;
- type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;
- type 3, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

ISO/TR 14906, which is a Technical Report of type 2, was prepared by European Committee for Standardization (CEN) in collaboration with ISO Technical Committee ISO/TC 204, *Transport information and control systems*, in accordance with the Agreement on technical cooperation between ISO and CEN (Vienna Agreement).

© ISO 1998

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

International Organization for Standardization
Case postale 56 • CH-1211 Genève 20 • Switzerland
Internet iso@iso.ch

Printed in Switzerland

This document is being issued in the Technical Report (type 2) series of publications (according to subclause G.3.2.2 of part 1 of the ISO/IEC Directives, 1995) as a “prospective standard for provisional application” in the field of *transport information and control systems* because there is an urgent need for guidance on how standards in this field should be used to meet an identified need.

This document is not to be regarded as an “International Standard”. It is proposed for provisional application so that information and experience of its use in practice may be gathered. Comments on the content of this document should be sent to the ISO/TC 204 Secretariat.

A review of this Technical Report (type 2) will be carried out not later than three years after its publication with the options of: extension for another three years; conversion into an International Standard; or withdrawal.

Annex A forms an integral part of this Technical Report. Annexes B to D are for information only.

CONTENTS

Foreword	vi
Introduction	vi
1 Scope	1
2 Normative references	1
3 Definitions	3
4 Abbreviations	5
5 EFC application interface architecture	6
5.1 Relation to the DSRC communication architecture	6
5.2 Usage of DSRC application layer by the EFC application interface.....	7
5.3 Addressing of EFC attributes	7
5.3.1 <i>Basic mechanism</i>	7
5.3.2 <i>Role of the EID</i>	8
5.3.3 <i>Multiple Instances of Attributes</i>	8
5.4 Addressing of components	9
6 EFC Transaction Model	10
6.1 Initialisation Phase	10
6.1.1 <i>Overview</i>	10
6.1.2 <i>EFC application-specific contents of the BST</i>	11
6.1.3 <i>EFC application-specific contents of the VST</i>	11
6.2 Transaction phase	12
7 EFC Functions	14
7.1 Overview and general concepts.....	14
7.1.1 <i>EFC functions and service primitives</i>	14
7.1.2 <i>Overview of EFC functions</i>	15
7.1.3 <i>Handling of multiple instances</i>	16
7.1.4 <i>Security</i>	17
7.2 EFC functions	18
7.2.1 <i>GET_STAMPED</i>	18
7.2.2 <i>SET_STAMPED</i>	19
7.2.3 <i>GET_SECURE</i>	20
7.2.4 <i>SET_SECURE</i>	21
7.2.5 <i>GET_INSTANCE</i>	22
7.2.6 <i>SET_INSTANCE</i>	23
7.2.7 <i>GET_NONCE</i>	24
7.2.8 <i>SET_NONCE</i>	25
7.2.9 <i>TRANSFER_CHANNEL</i>	26
7.2.10 <i>COPY</i>	27
7.2.11 <i>SET_MMI</i>	28
7.2.12 <i>SUBTRACT</i>	29
7.2.13 <i>ADD</i>	30
7.2.14 <i>DEBIT</i>	31
7.2.15 <i>CREDIT</i>	32
7.2.16 <i>ECHO</i>	33
8 EFC Attributes	34
8.1 Data group CONTRACT	35
8.2 Data group RECEIPT	36
8.3 Data group VEHICLE	38
8.4 Data group EQUIPMENT	39
8.5 Data group DRIVER.....	39
8.6 Data group PAYMENT	40

ANNEX A (normative) EFC data type specifications 41

Annex B (informative) An excerpt from DSRC application layer 48

B.1 Format of service primitives 48

B.2 Generic DSRC application layer functions 50

B.2.1 GET 50

B.2.2 SET 50

B.3 Container ASN.1 type definition 51

Annex C (informative) Examples of EFC transactions using the EFC application interface 52

C.1 Example of an EFC transaction - Example 1 52

C.2 Example of an EFC transaction - Example 2 53

C.3 Example of an EFC transaction - Example 3 54

C.4 Example of an EFC transaction - Example 4 55

C.5 Example of an EFC transaction - Example 5 58

ANNEX D (informative) Functional requirements 62

Foreword

This European Prestandard has been prepared by Technical Committee CEN/TC 278 "Road transport and traffic telematics", the secretariat of which is held by NNI, in collaboration with Technical Committee ISO/TC 204 "Transport information and control systems".

According to the CEN/CENELEC Internal Regulations, the national standards organizations of the following countries are bound to announce this European Prestandard: Austria, Belgium, Czech Republic, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and the United Kingdom.

Introduction

This European Pre-Standard specifies an application interface for Electronic Fee Collection (EFC) systems, which are based on the Dedicated Short-Range Communication (DSRC), enabling interoperability between open EFC systems (i.e. between different EFC system operators) on an EFC-DSRC application interface level.

The European Pre-Standard provides specifications for the EFC transaction model, EFC data elements (referred to as attributes) and functions, from which an EFC transaction can be built. The EFC transaction model provides a mechanism that allows handling of different versions of EFC transactions and associated contracts. A certain EFC transaction supports a certain set of EFC attributes and EFC functions as defined in this European Pre-Standard. It is not envisaged that the complete set of EFC attributes and functions is present in each piece of EFC equipment, be OBE or RSE.

This European Pre-Standard provides the basis for agreements between operators, which are needed to achieve interoperability. Based on the tools specified in this European Pre-Standard, interoperability can be reached by operators recognising each others EFC transactions (including the exchange of security algorithms and keys) and implementing the EFC transactions in each others RSE, or they may reach an agreement to define a new transaction (and contract) that is common to both. Considerations also have to be made by each operator that the RSE has sufficient resources to implement such additional EFC transactions.

In order to achieve interoperability, operators have to agree on issues like:

- which optional features are actually being implemented and used;
- security policy (including encryption algorithms and key management, if applicable);
- operational issues, such as how many receipts may be stored for privacy reasons, how many receipts are necessary for operational reasons (e.g. as entry tickets or as proof of payment);
- the agreements needed between operators in order to regulate the handling of different EFC transactions.

This European Pre-Standard has the following structure. In the first four clauses the scope, normative references, definitions of terms and abbreviations are accounted for. Next, in clause 5, the EFC Application interface architecture is described in terms of its relation to the DSRC communication architecture, including the addressing of data attributes and of components. In the following clause 6, the EFC transaction model is introduced, defining the common steps of each EFC transaction, in particular the initialisation phase. Clauses 7 and 8 are dedicated to the detailed specification of the EFC application functions and of the EFC data attributes, respectively. Four annexes provide

- the normative ASN.1 specifications of the used data types (EFC action parameters and attributes);
- an informative excerpt from DSRC application layer (ENV 12834), and its service primitives, parameters and functions;
- informative examples of EFC transactions using the specified EFC attributes and functions;
- an informative listing of functional requirements, which can be satisfied by using the tools provided by this European Pre-standard.

1 Scope

This European Pre-Standard specifies the application interface in the context of Electronic Fee Collection (EFC) systems using the Dedicated Short-Range Communications (DSRC).

The EFC application interface is the EFC application process interface to the DSRC Application Layer, as can be seen in figure 1 below. The scope of this European Pre-Standard comprises specifications of:

- EFC attributes (i.e. EFC application information);
- the addressing procedures of EFC attributes and (hardware) components (e.g. ICC and MMI);
- EFC application functions, i.e. further qualification of actions by definitions of the concerned services, assignment of associated ActionType values and content and meaning of action parameters;
- the EFC transaction model, which defines the common elements and steps of any EFC transaction;
- the behaviour of the interface so as to ensure interoperability on an EFC-DSRC application interface level.

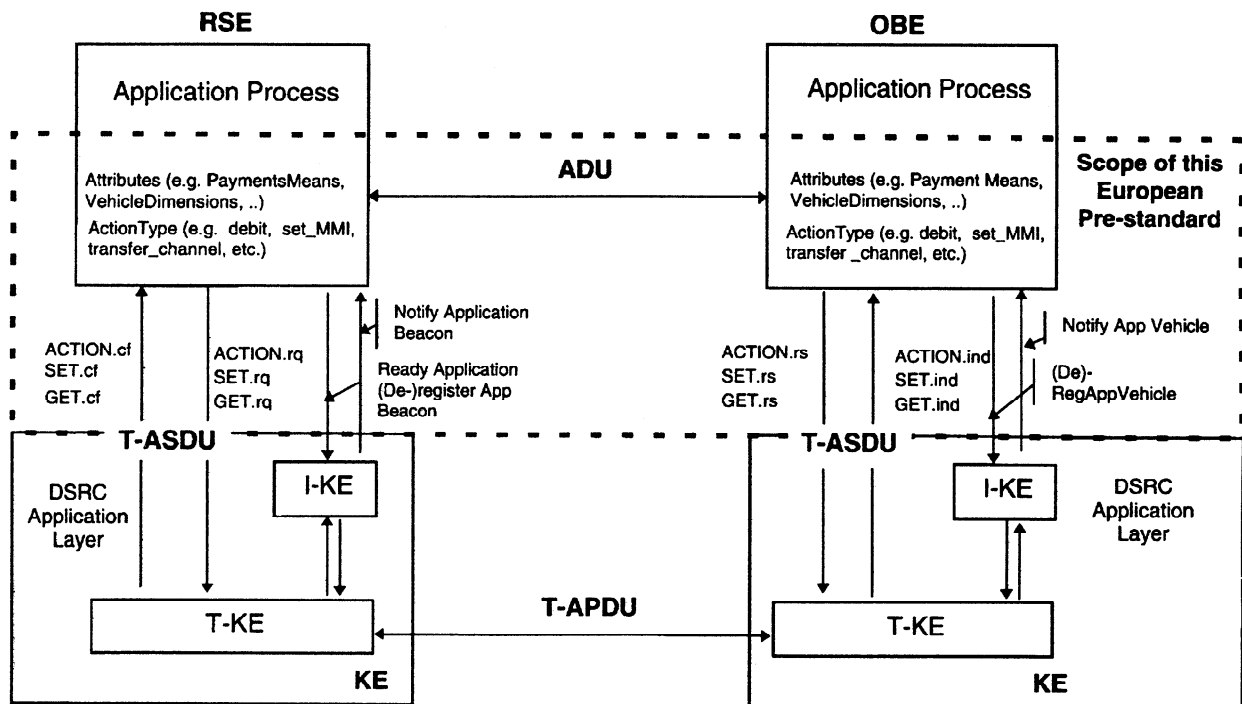


Figure 1: The EFC application interface

This is an interface standard, adhering to the open systems interconnection (OSI) philosophy (ISO/IEC 7498-1), and it is as such not concerned with the implementation choices to be realised at either side of the interface.

This European Pre-Standard provides security-specific functionality as place holders (data and functions) to enable the implementation of secure EFC transactions. Yet the specification of the security policy (including specific security algorithms and key management) remains at the discretion and under the control of the EFC operator, and hence is outside the scope of this European Pre-Standard.

2 Normative references

This European Pre-Standard incorporates by dated or undated reference, provisions from other publications.

These normative references are cited at the appropriate places in the text and the publications are listed hereafter.

For dated references, subsequent amendments to or revisions of any of these publications apply to this European Pre-Standard only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies.

ISO 612:		Vehicle Measurement Definition
ISO 1176:		Vehicle Weight Definitions
ISO 3166:		Codes for the representation of names of countries
ISO 3779:	1983	Road Vehicles - Vehicle Identification Number (VIN) Content and Structure
ISO 3780:	1983	Road Vehicles World manufacturer identification code (WMI)
ISO 4217:		Codes for the representation of currencies and funds
ISO 7498-1:	1994	Information Processing Systems - Open Systems Interconnection - Basic Reference model
ISO/IEC 8824-1:	1995	Information processing systems - Open Systems Interconnection - Specification of abstract syntax notation one (ASN.1)
ISO/IEC 8825-2:	1996	Information processing systems - Open Systems Interconnection - ASN.1 encoding rules: Specification of Packed encoding rules
ENV 1545-1	1998	Identification card systems - Surface transport applications - Part 1 : General data elements
ENV 1545-2	1998	Identification card systems - Surface transport applications - Part 2 : Transport payment related data elements
prENV ISO 14816:	1998	Road Traffic and Transport Telematics (RTTT), Automatic Vehicle and Equipment Identification - Numbering and Data Structures (ISO/DTR 14816 : 1998)
ENV 12834:	1997	Road Traffic and Transport Telematics (RTTT), Dedicated Short-Range Communication (DSRC) - Application Layer

3 Definitions

For the purposes of this European Pre-Standard, the following definitions apply:

- 3.1 access credentials** Data that is transferred to *On-Board Equipment*, in order to establish the claimed identity of an RSE application process entity.
- NOTE: The access credentials carries information needed to fulfil access conditions in order to perform the operation on the addressed element in the OBE. The access credentials can carry passwords as well as cryptographic based information such as authenticators.*
- 3.2 action** Function that an application process resident at the *Roadside Equipment* can invoke in order to make the *On-Board Equipment* execute a specific operation during the *Transaction*.
- 3.3 attribute** Application information formed by one or by a sequence of data elements, and is managed by different actions used for implementation of a *transaction*.
- 3.4 authenticator** Data appended to, or a cryptographic transformation (see cryptography) of, a data unit that allows a recipient of the data unit to prove the source and/or the integrity of the data unit and protect against forgery.
- 3.5 channel** An information transfer path [ISO/IEC 7498-2].
- 3.6 component** Logical and physical entity composing an *On-Board Equipment*, supporting a specific functionality.
- 3.7 contract** Expression of an agreement between two or more parties concerning the use of the road infrastructure.
- 3.8 cryptography** The discipline which embodies principles, means, and methods for the transformation of data in order to hide its information content, prevent its undetected modification or/and prevent its unauthorised use [ISO/IEC 7498-2].
- 3.9 data group** A collection of closely related EFC data attributes which together describe a distinct part of an Electronic Fee Collection transaction.
- 3.10 data integrity** The property that data has not been altered or destroyed in an unauthorised manner [ISO 7498-2].
- 3.11 element** In the context of DSRC, a directory containing application information in form of *Attributes*.
- 3.12 on-board equipment** Equipment located within the vehicle and supporting the information exchange with the *Road Side Equipment*. It is composed of the *On-Board Unit* and other sub-units whose presence have to be considered optional for the execution of a *Transaction*.
- 3.13 on-board unit** Minimum component of an *On-Board Equipment*, whose functionality always includes at least the support of the DSRC interface.
- 3.14 roadside equipment** Equipment located at a fixed position along the road transport network, for the purpose of communication and data exchanges with the *On-Board Equipment* of passing vehicles.

- 3.15 service (EFC)** Road transport related facility provided by a *Service Provider*. Normally a type of infrastructure, the use of which is offered to the *User* for which the *User* may be requested to pay.
- 3.16 service primitive (communication)** Elementary communication service provided by the Application layer protocol to the application processes.
- NOTE: The invocation of a service primitive by an application process implicitly calls upon and uses services offered by the lower protocol layers.*
- 3.17 service provider (EFC)** The operator that accepts the user's payment means and in return provides a road-use service to the user.
- 3.18 session** The complete exchange of information and interaction occurring at a specific **Electronic Fee Collection** station between the *Roadside Equipment* and the user/vehicle.
- 3.19 transaction** The whole of the exchange of information between the *Roadside Equipment* and the *On-Board Equipment* necessary for the completion of an **Electronic Fee Collection** operation over the DSRC.
- NOTE: A transaction may require more than one session in order to be achieved, e.g. an entry session and an exit session.*
- 3.20 transaction model** Functional model describing the general structure of Electronic Fee Collection transactions.
- 3.21 user** The entity that uses transport services provided by the *Service Provider* according to the terms of a *Contract*.

4 Abbreviations

For the purpose of this European Pre-Standard, the following abbreviations apply throughout the document unless otherwise specified:

ADU	Application Data Unit
APDU	Application Protocol Data Unit
AP	Application Process
ASN.1	Abstract Syntax Notation One
BST	Beacon Service Table (DSRC Application Layer)
DSRC	Dedicated Short-Range Communications
EID	Element ID
EFC	Electronic Fee Collection
GPS	Global Positioning System
ICC	Integrated Circuit(s) Card
I-KE	Initialisation Kernel Element (DSRC Application Layer)
IID	Invoker ID
LID	Link ID
MMI	Man-Machine Interface
OBE	On-Board Equipment
OBU	On-Board Unit
PDU	Protocol Data Unit
PER	Packed Encoding Rules
RSE	Road-Side Equipment
RTTT	Road Traffic and Transport Telematics
SAM	Secure Application Module
T-APDU	Transport-Application Protocol Data Unit (DSRC Application Layer)
T-ASDU	Transport-Application Service Data Unit (DSRC Application Layer)
T-KE	Transport Kernel Element (DSRC Application Layer)
VST	Vehicle Service Table (DSRC Application Layer)

5 EFC application interface architecture

5.1 Relation to the DSRC communication architecture

The DSRC services are provided to an application process by means of the DSRC Application Layer service primitives, which are abstract implementation interactions between a communication service user and provider. The services are offered by the DSRC communication entities by means of its DSRC Application Layer (ENV 12834).

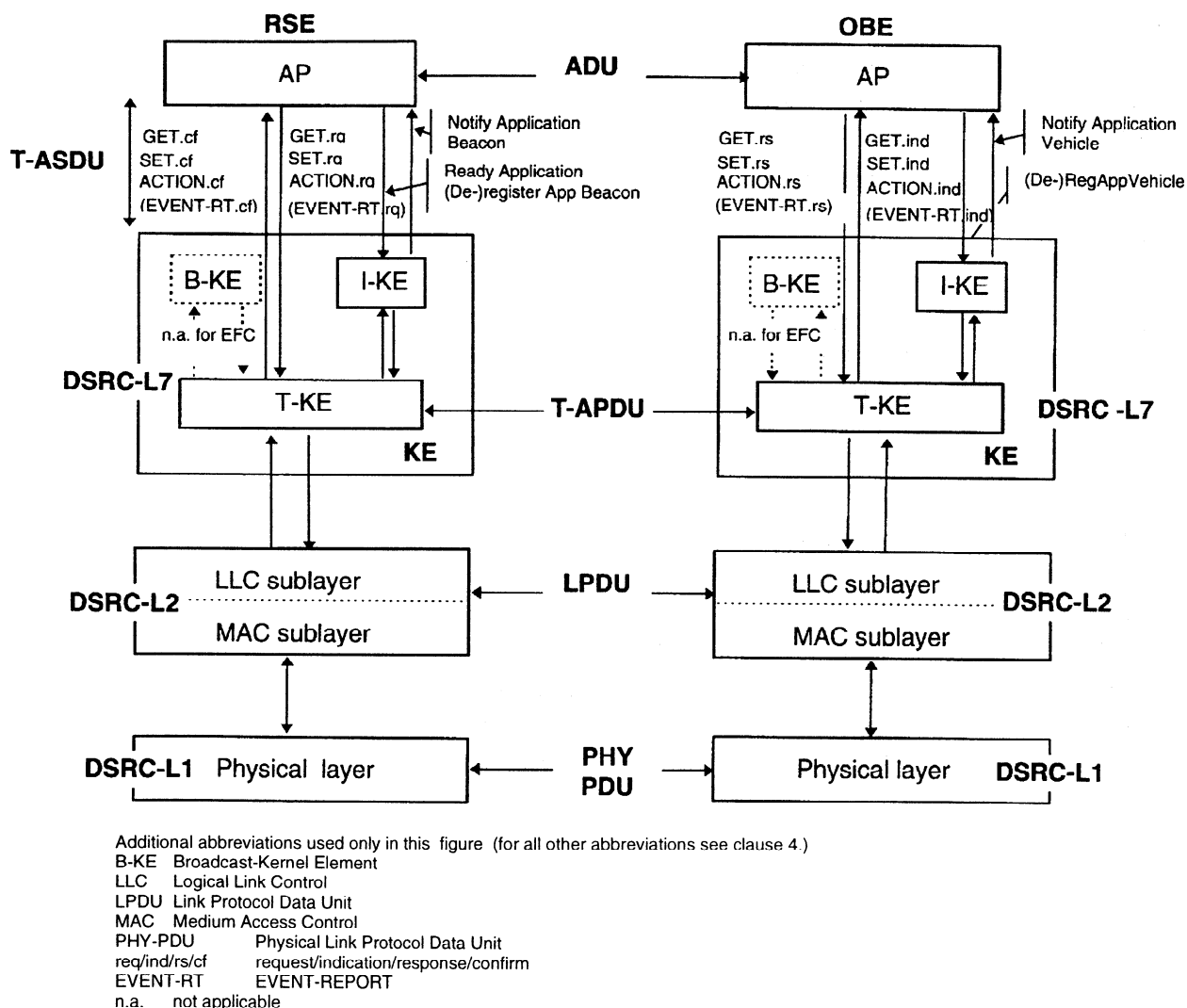


Figure 2: The EFC application process on top of the DSRC communication stack

The Transfer Kernel Element (T-KE) of DSRC Application Layer offers the following services to application processes (see also figure 2 above):

- **GET:** The invocation of a GET service request results in retrieval (i.e. a reading) of application information (i.e. Attributes) from the peer service user (i.e. the OBE application process), a reply is always expected.
- **SET:** The invocation of a SET service request results in modification (i.e. writing) of application information (i.e. Attributes) of the peer service user (i.e. the OBE application process). This service may be requested in confirmed or non-confirmed mode, a reply is only expected in the former case.

- **ACTION:** The invocation of an ACTION service request results in a performance of an action by the peer service user (i.e. the OBE application process). An action is further qualified (see clause 7.4) by the value of the ActionType. This service may be requested in confirmed or non-confirmed mode, a reply is only expected in the former case.
- **EVENT-REPORT:** The invocation of an EVENT-REPORT service request forwards a notification of an event to the peer service user.
- **INITIALISATION:** The invocation of an initialisation service request by RSE results in an attempt to initialise communication between a RSE and each OBE that has not yet established communication with the concerned RSE. The Initialisation service is only used by the I-KE as defined in ENV 12834, which in its turn is configured by the application(s) wishing to execute applications over the DSRC link.

5.2 Usage of DSRC application layer by the EFC application interface

EFC uses the following services offered by DSRC Application Layer (as defined in ENV 12834):

- The INITIALISATION services:
 - (De-)Register Application RSE (at RSE)
 - Notify Application Beacon (at RSE)
 - Ready Application (at RSE)
 - (De-)Register Application OBE (at OBE)
 - Notify Application Vehicle (at OBE)
 are used to realise the EFC-specific initialisation mechanism (see clause 6)
- The GET service is used to retrieve EFC attributes (see annex B.2.1. For attribute specifications see clause 8)
- The SET-service is used to set EFC attributes (see annex B.2.2)
- The ACTION-services are applied to realise additional EFC specific functionality needed to support EFC application processes, such as TRANSFER_CHANNEL, SET_MMI and ECHO (see clause 7.2).

In the following, the EFC-specific usage of the DSRC Layer 7 services is specified in detail.

NOTE: The EVENT-REPORT-service can be implicitly used by EFC application processes. It is e.g. used indirectly as part of an already defined command to release an application process (see Ready Application of ENV 12834). However as the EVENT-REPORT-service is not explicitly used by EFC application processes, this service is not further referred to in this document.

5.3 Addressing of EFC attributes

5.3.1 Basic mechanism

EFC Attributes are used to transfer the EFC application-specific information.

EFC Attributes are composed of one or more data elements of specified ASN.1 types. All data elements of an ASN.1 type are mandatory. To each data element an unambiguously defined name is associated.

To each EFC Attribute, an AttributeID is associated. The AttributeID enables to unambiguously identify and address an EFC Attribute.

EXAMPLE: Figure 3 illustrates the basic addressing mechanism:

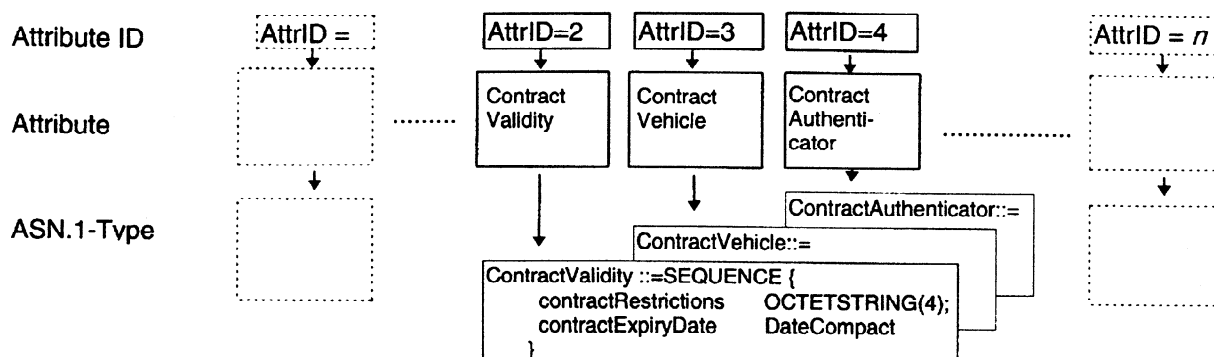


Figure 3: Basic addressing mechanism

5.3.2 Role of the EID

The DSRC-EID (different from 0) is used to identify an EFC context, given by the EFC-ContextMark (see clause 6.1.3), in which Attributes can be addressed unambiguously by AttributeIDs. In the VST, the OBE may specify several of these EFC contexts, each corresponding to a set of EFC Attributes and EFC functions supported by it.

EXAMPLE:

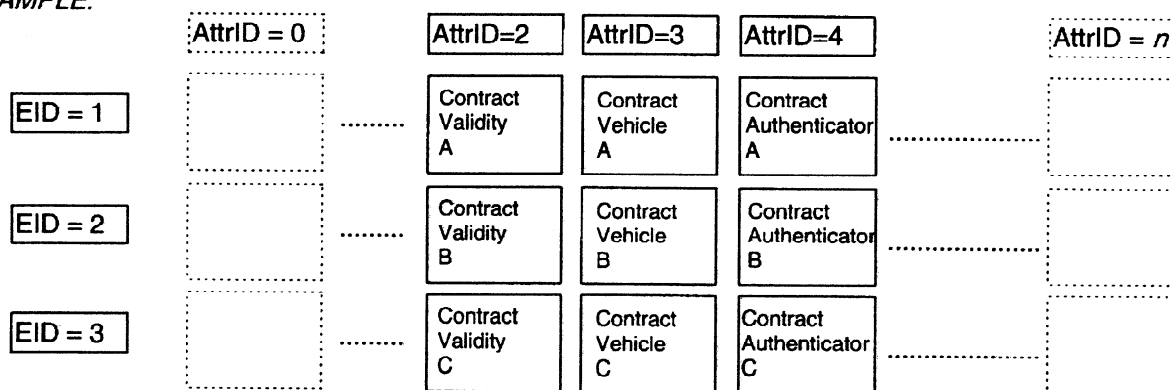


Figure 4: Role of the EID

EID equals 0 shall be used to address application-independent functions and components, e.g. SET_MMI and TRANSFER_CHANNEL (see clause 7.2).

5.3.3 Multiple Instances of Attributes

There may be n instances of an Attribute available in an OBE.

The maximum number of instances $N_{\max}$ of one Attribute may be limited according to the needs of operators and users. The default maximum number of instances is $N_{\max}=1$. The value of $N_{\max}$ is determined at the time of OBE configuration.

EXAMPLE:

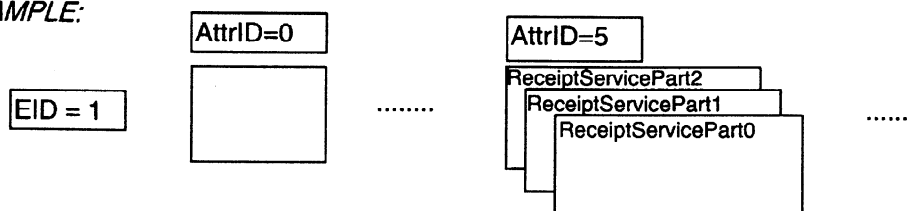


Figure 5: Multiple instances (0-2) of attribute 5

The handling of multiple instances and the corresponding addressing mechanism are described in detail as part of the behaviour specification of the corresponding functions supporting multiple instances (see clause 7.2.5 for GET_INSTANCE and clause 7.2.6 for SET_INSTANCE).

5.4 Addressing of components

Components of an OBE to be addressed via the EFC Application Interface include for example:

- OBU
- SAM 1
- SAM 2
- ICC
- Display
- Beeper
- Printer
- Serial interface
- Parallel interface
- GPS
- Tachograph

Addressing of these components is enabled on two levels, device-specific and device-independent addressing.

The **device-specific transparent addressing mechanism** enables the transfer of information, which shall be processed by the addressed device (such as an ICC-command). The addressed device is identified by a *channel Id*. The EFC function TRANSFER_CHANNEL (see clause 7.2.9) supports this functionality.

EXAMPLE: Transfer of a bit string to an ICC.

The **device-independent addressing mechanism** uses a set of commands, which describe a certain functionality, which can be performed by various OBU components. In this case, the operating system of the OBU will address the corresponding components. The EFC function SET_MMI supports this functionality (see clause 7.2.11).

EXAMPLE: Invocation of a SET_MMI(EID=0, ContactOperator) function activates an OBE MMI-device, e.g. a beeper or a display.

NOTE: In addition, the OBE may use a kind of implicit addressing. In an implementation, specific attributes or data elements may activate some MMI function (e.g. a SET command on the attribute ReceiptText might display the text on an LCD display. A SET command on the attribute ReceiptServicePart with data element SessionResultOperational other than SessionOK might activate an alert beep).

6 EFC Transaction Model

The EFC Transaction Model related to the EFC Application Interface for the DSRC comprises two phases, the initialisation phase and the transaction phase.

NOTE: The purpose of the initialisation phase is to set up the communication between the RSE and OBEs that have entered the DSRC zone but have not yet established communication with the RSE, and to notify the application processes. It provides amongst others a multi-application switching mechanism, allowing for execution of several RTTT applications (in parallel) at one RSE station.

The transaction phase can only be reached after completion of the initialisation phase. The EFC functions, as defined in clause 7, can be performed in the transaction phase.

6.1 Initialisation Phase

6.1.1 Overview

This clause provides an overview of the functionality of, and the information exchanges in, the initialisation phase.

The Initialisation procedures, by means of BST and VST exchanges, are defined in ENV 12834. Clauses 6.1.2 and 6.1.3 below account for the EFC application-specific information that shall be included in the BST and VST, respectively.

NOTE 1: The OBE evaluates the received BST, and selects the applications that it wishes to perform out of the lists of applications supported by the RSE. If the OBE does not support any of application(s) supported by the RSE, then the OBE shall not exchange any information with the RSE. If the OBE supports at least one of the application(s) supported by the RSE, then the OBE shall send its corresponding VST, informing the RSE of which application it wishes to execute.

NOTE 2: Figure 6 describes the initialisation phase.

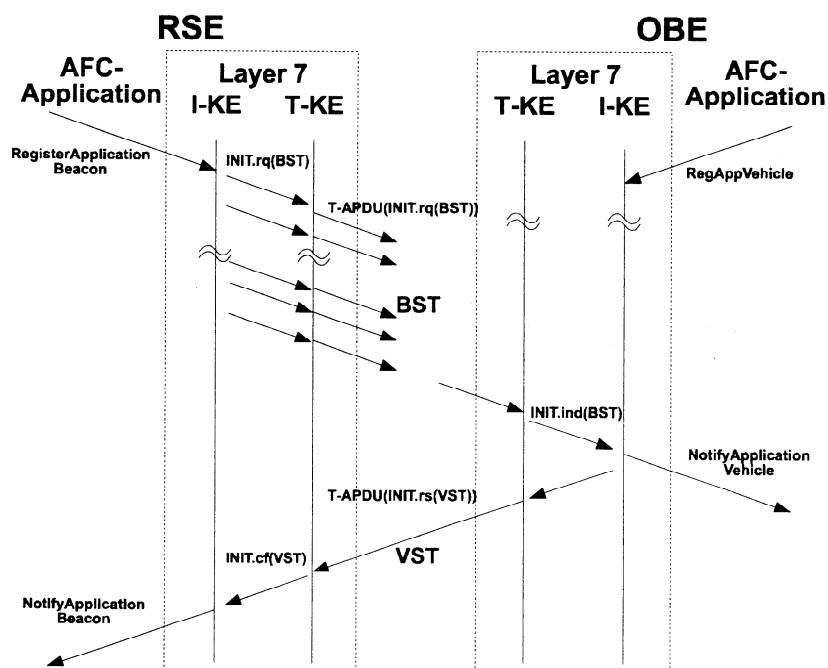


Figure 6: Initialisation phase: BST - VST exchanges

The Initialisation service associated with the initialisation phase is only used by the I-KE (of ENV 12834), which in its turn is configured by the application(s) wishing to execute applications over the DSRC link. The I-KEs of the RSE and of the concerned OBE shall have been configured, according to ENV 12834, prior to the invocation of the Initialisation service by the RSE.

6.1.2 EFC application-specific contents of the BST

An RSE supporting EFC shall have configured its I-KE to carry the following information related specifically to the EFC application(s):

- the application identifier (AID) shall be equal to 1 (i.e. the value assigned for EFC);
- the EFC application shall be qualified as a mandatory application;
- EID shall not be transmitted in the BST related to the EFC application;
- No Parameter shall be transmitted in the BST related to the EFC application.

There shall be only one EFC application present in the BST (i.e. there shall be only one instance of aid=1 in the BST) regardless of whether the RSE supports more than one EFC-ContextMark (see also clause 6.1.3) or not.

NOTE 1: The above is the EFC application-specific contents of the BST. The complete BST is defined in ENV 12834 and is given below for readability:

```
BST ::= SEQUENCE {
    beacon BeaconID,
    time Time,
    profile Profile,
    mandApplications ApplicationList,
    nonmandApplications ApplicationList OPTIONAL,
    profileList SEQUENCE(0...127, ...) OF Profile
}
```

where:

```
ApplicationList ::= SEQUENCE (0..127,...) OF
SEQUENCE{
    aid DSRCApplicationEntityID, --aid=1
    eid Dsrc-EID OPTIONAL, --empty
    parameter Container OPTIONAL --empty
}
```

NOTE 2: There is likely to be assigned other aid values for identification of non-European EFC contexts, where those contexts will define the usage of eid and parameter fields (e.g. non empty).

6.1.3 EFC application-specific contents of the VST

Each EFC application and corresponding contract shall be associated with an EFC-ContextMark, as defined below. An OBE may support several EFC applications. If several EFC applications are supported by an OBE, then the order in which the EFC-ContextMark are presented in the VST shall correspond to the user's order of preference. The RSE should honour the first EFC-ContextMark that it supports as presented in the VST.

An OBE supporting EFC shall have configured its I-KE to carry the following information related specifically to the concerned EFC application:

- the AID shall be equal to 1;
- the EID value shall be unique within the OBE throughout the complete DSRC session, and shall be logically associated with the corresponding EFC-ContextMark contained in the Parameter;
- the Parameter shall be of Container CHOICE type OCTET STRING and shall comprise the EFC-ContextMark as defined below, and may also be configured to carry additional EFC attributes (as defined in clause 8 and annex A).

EFC-ContextMark::=SEQUENCE{

ContractProvider	Provider,
TypeOfContract	OCTET STRING (SIZE(2)),
ContextVersion	INTEGER(0..127,...)
}	

The **EFC-ContextMark** denotes a specific EFC context in the OBE, comprising the organisation that issued the contract, the type of contract and the context version. ContractProvider, TypeOfContract and ContextVersion are further defined in clause 8 as data elements of the Attribute EFC-ContextMark.

NOTE 1: The above is the EFC application-specific contents of the VST. The complete VST is defined in ENV 12834 and is given below for readability:

```

VST ::= SEQUENCE {
    fill          Bit STRING (SIZE(4))
    profile       Profile,
    applications  ApplicationList,
    obeConfiguration ObeConfiguration
}

where:
ApplicationList ::= SEQUENCE (0..127,...) OF
SEQUENCE{
    aid          DsrcApplicationEntityID,    --aid =1
    eid          Dsrc-EID    OPTIONAL,      --eid = e.g. 2
    parameter    Container   OPTIONAL      --EFC-ContextMark
                                           --plus any EFC Attribute
}

```

NOTE 2: There is likely to be assigned other aid values for identification of non-European EFC contexts, where those contexts will define the usage of eid and parameter fields.

NOTE 3: An EFC application provider retains the ultimate control of his security domain, i.e. the security level and the associated security mechanisms to be used within his system.

6.2 Transaction phase

After completion of the Initialisation phase, the appropriate RSE application is informed (by means of the Notify Application Beacon service) of the EFC-ContextMark(s) and associated EID(s). The RSE shall use the functions defined in clause 7 to complete the EFC transaction.

The RSE may invoke any sequence of EFC functions to complete the EFC transaction, provided that they are supported by the EFC-ContextMark. The OBE shall respond to the EFC functions invoked by the RSE, and shall not initiate any EFC functions (by usage of a request service primitive, see further clause 7) on its side.

EXAMPLE: A transaction may consist of the following steps:

- GET(EID, ContractValidity, ContractVehicle, ReceiptServicePart, PaymentMeansBalance)
- DEBIT(EID, Fee)
- SET(EID, ReceiptServicePart)

Due to the construction of the EFC part of the VST, each EID identifies a certain EFC-ContextMark and shall be used by the RSE as parameter of every function to unambiguously address data elements within the context given by the EFC-ContextMark. More than one EID may be used in one session.

Both which attributes are implemented and which are not, and the maximum number of instances of an attribute is defined at time of configuration of the OBE and is outside the scope of this European Pre-Standard. These implementation dependent aspects are referenced unambiguously by the ContextVersion data element in the EFC-ContextMark.

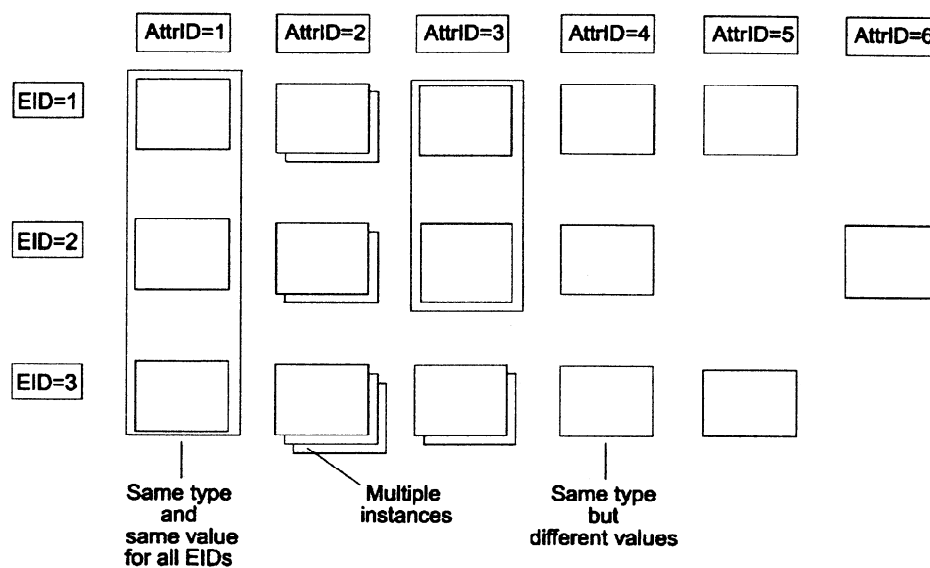


Figure 7: Context of attributes given by EFC-ContextMark and identified by the EID

NOTE 1: This construction of contexts being identified by EIDs allows amongst others to implement the following transactions:

- booking from two contracts in one transaction. There is sometimes the need to book from two contracts in one session, e.g. when a customer has a contract with a reduced price (e.g. a commuter contract) for part of the route being tolled, plus a standard (not reduced) contract for the rest of the route. This may be implemented by having all data groups identical between two EIDs, except for the data group contract.
- having either two instances of the data group Vehicle to accommodate a pulling vehicle plus a trailer, or having two EIDs with separate data groups Vehicle for pulling vehicle and trailer (and probably also separate Contract)

NOTE 2: This EFC transaction model and associated procedures allow for different levels of co-existence and interoperability between operators:

- No agreement between operators - each operator has a completely separate application domain, i.e. there are no common data groups. Each operator books by using "his" EID.
- Agreement to share some data groups, but not others. E.g. the data groups Vehicle, Receipt and Payment are shared, but not Contract. Different security measures (algorithms) are used by the two operators. Or, all data groups are shared, except for Payment - each operator books from an account issued by himself.

7 EFC Functions

7.1 Overview and general concepts

7.1.1 EFC functions and service primitives

This clause describes the EFC functions invoked by T-ASDUs exchanged between peer applications communicating via a DSRC link. The T-ASDUs are exchanged by means of service primitives of the DSRC Application Layer (ENV 12834). Exchanges of service primitives (and the corresponding T-ASDUs) associated with EFC functions adhere to the following basic pattern:

- xxx.rq (request) service primitive invoked by the RSE application to DSRC Application Layer;
- xxx.ind (indication) service primitive issued by the DSRC Application Layer to the OBE application;
- xxx.rs (response) service primitive invoked by the OBE application to DSRC Application Layer;
- xxx.cf (confirmation) service primitive issued by the DSRC Application Layer to RSE application.

The last two steps are either mandatory or optional, depending on the nature of the service primitive and on the setting of the Mode parameter (see clause 6.2.2.4 in ENV 12834, or annex B.1 in this European Pre-Standard).

The logical sequence of a successful service primitives exchange (for Mode = TRUE) is illustrated in figure 8 below. Service primitives that occur earlier in time, and are connected by dotted lines in the figure, are the logical antecedents of subsequent service primitives.

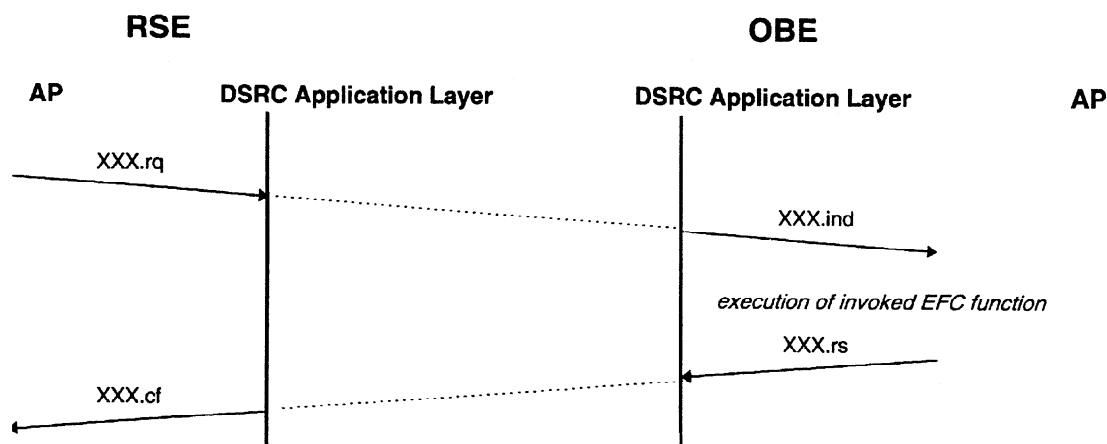


Figure 8: The logical sequencing of service primitive exchanges

For the purposes of this European Pre-Standard, the DSRC link is seen as completely transparent, i.e. in the absence of exceptions the XXX.ind is identical in content and meaning to the XXX.rq, and the XXX.cf is identical in content and meaning to the XXX.rs. For the purpose of conciseness there will be:

- one description for *request*, i.e. XXX.rq, covering both request and indication service primitives;
- one description for *response*, i.e. XXX.rs, covering both response and confirmation service primitives.

The format and the parameters of the service primitives of the DSRC application layer are defined in ENV 12834, clause 6.2.2 (T-KE Services).

NOTE: Annex B.1 accounts for the format and the definitions of the parameters of the service primitives in form of an excerpt from ENV 12834 in order to facilitate the reading of this European Pre-Standard.

7.1.2 Overview of EFC functions

This clause provides an overview of the EFC functions (based on the Action service primitive of ENV 12834) that are defined in clause 7.2 of this European Pre-Standard. Each EFC function comprises a pair of service primitives, a request and its associated response service primitive, which are accounted for in clauses 7.2 but not in table 1 below.

Table 1: Overview of EFC functions

Function Name	Action Type	Action Parameter	Response Parameter	Remarks
GET_STAMPED	0	GetStampedRq	GetStampedRs	retrieves data with an authenticator from the OBE
SET_STAMPED	1	SetStampedRq	OCTET STRING	sets data in the OBE, which generates an authenticator
GET_SECURE	2	OCTET STRING	OCTET STRING	gets data securely from the OBE
SET_SECURE	3	OCTET STRING	OCTET STRING	sets data securely in the OBE
GET_INSTANCE	4	GetInstanceRq	GetInstanceRs	retrieves a number of entries out of an attribute's multiple instances
SET_INSTANCE	5	SetInstanceRq	n.a.	sets one entry at a specified position in an attribute's multiple instances
GET_NONCE	6	n.a.	OCTET STRING	retrieves a nonce - typically used against replay attacks
SET_NONCE	7	OCTET STRING	n.a.	sets a nonce - typically used against replay attacks
TRANSFER_CHANNEL	8	ChannelRq	ChannelRs	sets and/or retrieves data from the addressed OBE component (e.g. ICC)
COPY	9	CopyRq	n.a.	Copies data from a source EID to a destination EID
SET MMI	10	SetMMIRq	n.a.	invokes an MMI function (e.g. signal Ok via buzzer)
SUBTRACT	11	SubRq	n.a.	subtracts the given value to the addressed value
ADD	12	AddRq	n.a.	adds the given value to the addressed value
DEBIT	13	DebitRq	DebitRs	debits purse
CREDIT	14	CreditRq	CreditRs	credits purse
ECHO	15	OCTET STRING	OCTET STRING	OBE echoes received data
	16-31	Container	Container	future CEN EFC use

The GET and SET services (DSRC application layer functions) as defined in ENV 12834 (in clause 6.2) may also be used in an EFC transaction phase.

NOTE 1: GET is used to retrieve (i.e. read) value(s) of the addressed attribute(s), a reply is always expected. SET is used to set (i.e. write) value(s) of the addressed attribute(s). Annex B.1 this European Pre-Standard accounts for the format and the parameters of the GET and SET service primitives, whilst annex B.2 accounts for the behaviour of the corresponding functions in order to facilitate the reading of this European Pre-Standard.

NOTE 2: ReadyApplication initiates the EVENT-REPORT(Release) function.

7.1.3 Handling of multiple instances

For the purpose of description, the number of instances is denoted by n . In general the EFC functions operating on multiple instances of OBE Attributes can be divided into the following groups:

- GET, GET_STAMPED

These functions shall always access the last instance (i.e. instance at position 0). If no instance is available, the result is undefined but may lead to the return of an error code.

- GET_INSTANCE

This function carries parameters $N1$ and $N2$, both ≥ 0 . It shall return the following:

- if $N2 < N1$, or $N1 > n$, then the empty list;
- if $N2 \geq N1$ then the values of the instances numbered $N1$, $N1+1$, ... up to and including $\min(N2, n)$

NOTE: The case that zero instances are returned is legal, in this case the response carries an empty list.

- SET, SET_STAMPED

These functions shall always set the value of instance at position 0. In addition, the previous instance number p (where p is an integer between 0 and $N_{\max}-1$) shall become instance number $p+1$, and instance number $N_{\max}$ shall no longer be available.

NOTE 1: A cyclic buffer is as acceptable as a dynamic memory allocation scheme.

NOTE 2: The description above also covers the common case for $N_{\max} = 1$. In this case it leads to overwriting the old value of the single instance.

- SET_INSTANCE

This function carries a parameter $N \geq 0$, and a value for the addressed attribute. It shall always set the value of instance number N .

EXAMPLE 1: Behaviour for a static memory allocation scheme - a cyclic buffer
Assume $N_{\max} = 3$. Table 2 shows the effects of a certain sequence of functions.

Table 2: Behaviour for a static memory allocation scheme

Function	n	buffer content instance position			result
		0	1	2	
GET	3	X	X	X	returns X
GET_INSTANCE(1,0)	3	X	X	X	returns empty list
SET(A)	3	A	X	X	
GET	3	A	X	X	returns value A
SET(B)	3	B	A	X	
GET	3	B	A	X	returns value B
GET_INSTANCE(0,7)	3	B	A	X	returns list (B,A,X)
SET(C)	3	C	B	A	
GET	3	C	B	A	returns value C
GET_INSTANCE(1,2)	3	C	B	A	returns list (B, A)
SET(D)	3	D	C	B	value A is no longer available
SET_INSTANCE(1,E)	3	D	E	B	value C is no longer available

EXAMPLE 2: Behaviour for a dynamic memory allocation scheme
Assume $N_{\max} = 3$. Let $n = 0$ initially. Table 3 shows the effects of a certain sequence of functions.

Table 3: Behaviour for a dynamic memory allocation scheme

Function	n	buffer content instance position			result
		0	1	2	
GET	0				undefined (implementation dependent)
GET_INSTANCE(1,0)	0				returns empty list
SET(A)	1	<u>A</u>			
GET	1	<u>A</u>			returns value A
SET(B)	2	<u>B</u>	A		
GET	2	<u>B</u>	A		returns value B
GET_INSTANCE(0,6)	2	<u>B</u>	A		returns list(B, A)
SET(C)	3	<u>C</u>	B	A	
GET	3	<u>C</u>	B	A	returns value C
GET_INSTANCE(1,2)	3	<u>C</u>	B	A	returns list (B, A)
SET(D)	3	<u>D</u>	C	B	value A is no longer available
SET_INSTANCE(1,E)	3	<u>D</u>	E	B	value C is no longer available

7.1.4 Security

Whilst security is an essential part of EFC applications, the actual mechanisms are outside the scope of this European Pre-Standard. It is generally recognised that security mechanisms involve many parameters, like encryption algorithm and keys (if the security mechanism is encryption based at all), hash function, key length, padding method, redundancy data etc. It is assumed that the EFC application communicating parties know everything they need, either by implementation or by deriving information from the VST. This information should suffice for the RSE to determine how to proceed. The OBE, in general, supports only a limited number of security mechanisms.

In this European Pre-Standard, only a framework is defined permitting security mechanisms to be specified unambiguously at the discretion of the application provider. It includes the Access Credentials defined in ENV12834. In addition security values, like authenticators, will often be needed for additional protection.

7.1.4.1 Use of access credentials and authenticators

Access Credentials and Authenticators are defined as being of ASN.1 type OCTET STRING. This only pertains to the ASN.1 syntax, the semantics are implicit in the context given by the EFC-Context Mark as specified in the VST, and as selected by the EID.

Access Credentials shall be used to manage access to attributes. Different access conditions can apply for different attributes, and if so different access credentials should be associated with these access conditions.

EXAMPLE: The *VehicleDimensions* EFC attribute may be associated with no access conditions whilst the *ContractSerialNumber* and *ContractValidity* EFC attributes may be subject to access conditions (e.g. requiring the correct password to be presented).

Authenticators shall primarily pertain to values, and prove the source and/or the integrity of the data unit and protect against forgery. Authenticators are used in cryptography related EFC functions such as GET_STAMPED and SET_STAMPED etc. Authenticators can be transmitted from the RSE to the OBE, as Access Credentials in order to prove the authenticity of the RSE, or from the OBE to the RSE to prove the source and/or integrity of the data unit.

The security mechanisms to be applied, and the exact role of Access Credentials and Authenticators, will be determined by the owner of the EFC-ContextMark and is outside the scope of this European Pre-standard.