TECHNICAL SPECIFICATION

ISO/TS 14907-2

First edition
2006-06-15

Road transport and traffic telematics — Electronic fee collection — Test procedures for user and fixed equipment —

Part 2: Conformance test for the onboard unit application interface

Télématique de la circulation et du transport routier — Perception du télépéage — Modes opératoires relatifs aux équipements embarqués et aux équipements fixes —

Partie 2: Essai de conformité de l'interface d'application de l'unité embarquée

Reference number
ISO/TS 14907-2:2006(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

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 14907-2 was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

ISO/TS 14907 consists of the following parts, under the general title *Road transport and traffic telematics — Electronic fee collection — Test procedures for user and fixed equipment:*

— *Part 1: Description of test procedures*

— *Part 2: Conformance test for the onboard unit application interface*

Introduction

This CEN/ISO Technical Specification describes tests that verify OBU conformance of implementations of functions and data structures for EFC applications.

Road transport and traffic telematics — Electronic fee collection — Test procedures for user and fixed equipment —

Part 2:
Conformance test for the onboard unit application interface

1 Scope

This CEN/ISO Technical Specification describes tests that verify OBU conformance of implementations of functions and data structures, as defined in the implementation conformance statement based on ISO 14906, for EFC applications. After the tests of isolated data items and functions (C.1-C.2), an example is given for testing of a complete EFC transaction (C.3).

The scope of this CEN/ISO Technical Specification comprises definitions of OBU conformance assessment tests of:

— Basic DSRC L7 functionality;

— EFC application functions;

— EFC attributes (i.e. EFC application information);

— the addressing procedures of EFC attributes and (hardware) components (e.g. ICC and MMI);

— the EFC transaction model, which defines the common elements and steps of any EFC transaction; and

— the behaviour of the interface so as to support interoperability on an EFC-DSRC application interface level.

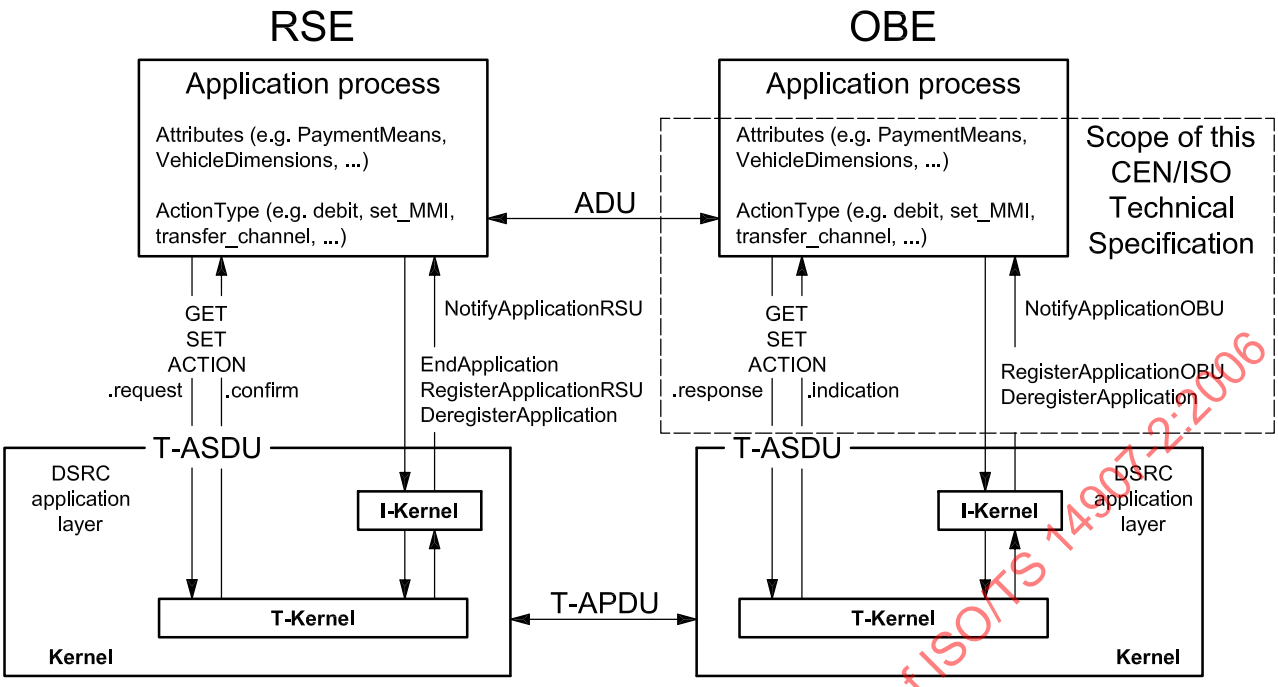

Figure 1 — The EFC application interface

The purpose of this CEN/ISO Technical Specification is to define tests that:

— assess OBU capabilities;

— assess OBU behaviour;

— serve as a guide for OBU conformance evaluation and type approval;

— achieve comparability between the results of the corresponding tests applied in different places at different times; and

— facilitate communications between parties.

Whereas this Technical Specification defines examples of test cases for DSRC and EFC functionality in Annex C, it does not intend to specify a complete test suite for a certain implementation. To compose a test suite for a specific EFC implementation, the test cases may have to be modified and new test cases may have to be defined and added in order for the conformance test to be complete. It may be useful to take into account the following considerations when defining a complete test suite:

— Small range: "exhaustive testing" of critical interoperability/compatibility features;

— Large range: testing of boundaries and random values;

— Composite types: testing of individual items in sequence or parallel.

Figure 2 shows the overall procedure of conformance testing.

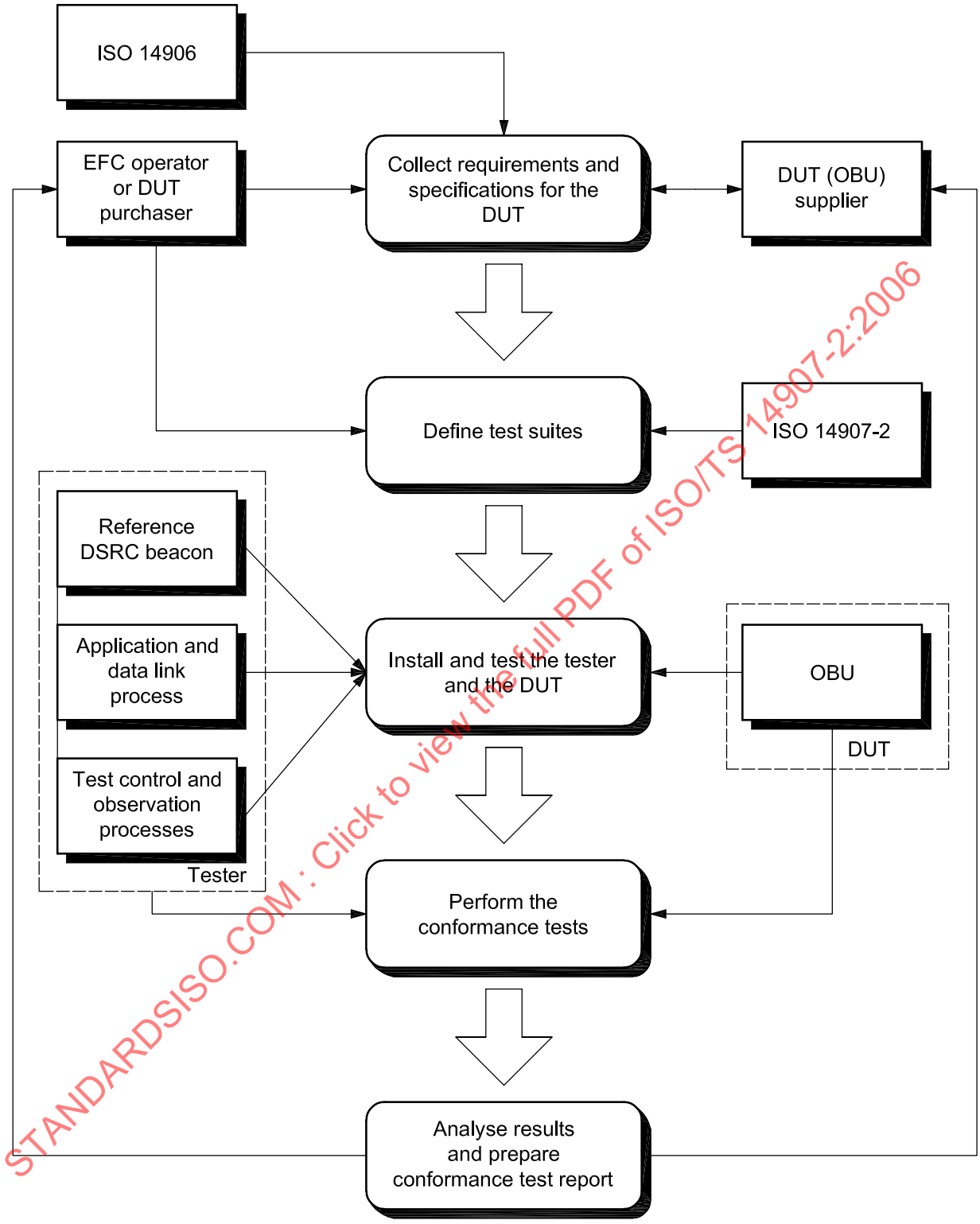

Figure 2 — Conformance testing process

Figure 3 gives a more detailed picture of the interface between the entity performing the conformance test and the supplier of the Device Under Test (DUT). By the EFC application specification, the implementation conformance statement pro forma and the implementation extra information for testing pro forma the supplier is requested to provide the DUT (OBU), containing the Implementation Under Test (IUT), as well as the documentation needed to perform the tests. More details on the content of the different documents are given in Clause 5 on OBU and supporting information.

NOTE 1 The Device Under Test contains the Implementation Under Test.

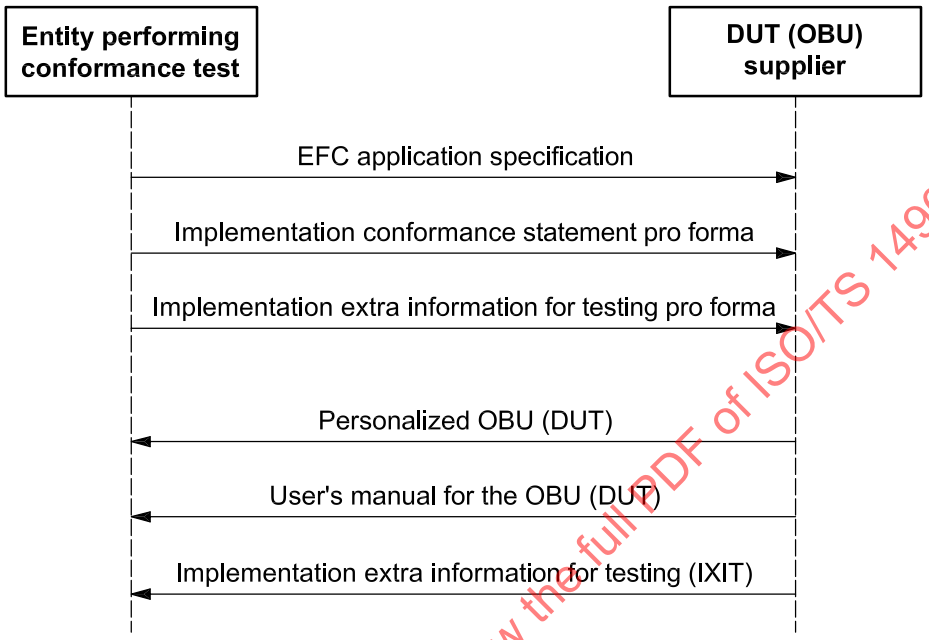

Figure 3 — Documentation DUT supplier

It is outside the scope of this CEN/ISO Technical Specification to define tests that assess:

— performance;

— robustness; and

— reliability of an implementation.

NOTE 2 ISO/TS 14907-1 defines test procedures that are aimed at assessing performance, robustness and reliability of EFC equipment and systems.

NOTE 3 The ISO/IEC 10373 family of International Standards defines test methods for proximity, vicinity, integrated circuits(s) cards and related devices that may be relevant for OBUs that support such cards.

2 Normative references

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

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

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

ISO/IEC 9646-1, *Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 1: General concepts*

ISO 14906:2004, *Road transport and traffic telematics — Electronic fee collection — Application interface definition for dedicated short-range communication*

ISO/TS 14907-1, *Road transport and traffic telematics — Electronic fee collection — Test procedures for user and fixed equipment — Part 1: Description of test procedures*

ISO/TS 17574, *Road transport and traffic telematics — Electronic fee collection (EFC) — Guidelines for EFC security protection profiles*

EN 12253, *Road transport and traffic telematics — Dedicated short-range communication — Physical layer using microwave at 5,8 GHz*

EN 12795, *Road transport and traffic telematics — Dedicated short-range communication (DRSC) — DRSC data link layer: medium access and logical link control*

EN 12834, *Road transport and traffic telematics — Dedicated short-range communication (DRSC) — DRSC application layer*

EN 13372, *Road transport and traffic telematics (RTTT) — Dedicated short-range communication — Profiles for RTTT applications*

3 Terms and definitions

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

3.1
access credentials
data that is transferred to OBE in order to establish the claimed identity of an RSE application process entity

[ISO 14906]

NOTE The access credentials carry 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 RSE can invoke in order to make the OBE execute a specific operation during the transaction

[ISO 14906]

3.3
attribute
application information formed by one or by a sequence of data elements, which is managed by different actions used for implementation of a transaction

[ISO 14906]

3.4
authenticator
data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and/or the integrity of the data unit and protect against forgery

[ISO 14906]

3.5
channel
information transfer path

[ISO/IEC 7498-2 and ISO 14906]

3.6
component
logical and physical entity composing OBE, supporting a specific functionality

[ISO 14906]

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

[ISO 14906]

3.8
cryptography
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 unauthorized use

[ISO/IEC 7498-2 and ISO 14906]

3.9
data group
collection of closely related EFC data attributes which together describe a distinct part of an EFC transaction

[ISO 14906]

3.10
data integrity
property that data has not been altered or destroyed in an unauthorized manner

[ISO/IEC 7498-2 and ISO 14906]

3.11
element
in the context of DSRC, a directory containing application information in form of attributes

[ISO 14906]

3.12
implementation conformance statement
statement made by the supplier of an implementation or system claimed to conform to a given specification, stating which capabilities have been implemented

3.13
implementation conformance statement pro forma
document, in the form of a questionnaire, which when completed for an implementation or system becomes an implementation conformance statement

3.14
implementation extra information for testing
statement made by the supplier or an implementer of an IUT which contains or references all of the information (in addition to that given in the implementation conformance statement) related to the IUT and its testing environment, which will enable the test laboratory to run an appropriate test suite against the IUT

3.15
implementation extra information for testing pro forma
document, in the form of a questionnaire, which when completed for an IUT becomes an implementation extra information for testing

3.16
on-board equipment
OBE
equipment located within the vehicle and supporting the information exchange with the RSE, it is composed of the OBU and other sub-units whose presence are considered optional for the execution of a transaction

[ISO 14906]

3.17
on-board unit
OBU
minimum component of an OBE, whose functionality always includes at least the support of the DSRC interface

[ISO 14906]

3.18
roadside equipment
RSE
equipment located at a fixed position along the road transport network, for the purpose of communication and data exchanges with the OBE of passing vehicles

[ISO 14906]

3.19
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

[ISO 14906]

3.20
service primitive (communication)
elementary communication service provided by the application layer protocol to the application processes

[ISO 14906]

NOTE The invocation of a service primitive by an application process implicitly calls upon and uses services offered by the lower protocol layers.

3.21
service provider (EFC)
operator that accepts the user's payment means and in return provides a road-use service to the user

[ISO 14906]

3.22
session
exchange of information and interaction occurring at a specific EFC station between the RSE and the user/vehicle

[ISO 14906]

3.23
transaction
whole of the exchange of information between the RSE and the OBE necessary for the completion of an EFC operation over the DSRC

[ISO 14906]

3.24
transaction model
functional model describing the general structure of EFC transactions

[ISO 14906]

3.25
tester
combination of equipment and processes which is able to perform conformance tests according to this Technical Specification

3.26
user
entity that uses transport services provided by the service provider according to the terms of a contract

[ISO 14906]

4 Abbreviations

For the purposes of this Technical Specification, the following abbreviations apply throughout the document unless otherwise specified.

4.1
ADU
Application Data Unit

4.2
APDU
Application Protocol Data Unit

4.3
AP
Application Process

4.4
ARIB
Association of Radio Industries and Businesses

NOTE ARIB (www.arib.or.jp) is based in Japan.

4.5
ASCII
American Standard Code for Information Interchange

4.6
ASN.1
Abstract Syntax Notation One

[ISO/IEC 8824-1]

4.7
ASP
Abstract Service Primitives

[ISO 9646-1]

4.8
AVI
Automatic Vehicle Identification

4.9
B-Kernel
Broadcast Kernel

4.10
BST
Beacon Service Table

4.11
cf
confirm

4.12
DSRC
Dedicated Short-Range Communication

4.13
DUT
Device Under Test

4.14
EID
Element Identifier

4.15
EFC
Electronic Fee Collection

4.16
FTP
File Transfer Protocol

4.17
HDLC
High-level Data Link Control

4.18
ICS
Implementation Conformance Statement

4.19
I-Kernel
Initialization Kernel

4.20
IID
Invoker Identifier

4.21
ind
indication

4.22
IUT
Implementation Under Test

[ISO 9646-1]

4.23
IXIT
Implementation eXtra Information for Testing

4.24
L1
Layer 1 of DSRC (physical layer)

4.25
L2
Layer 2 of DSRC (data link layer)

4.26
L7
Application Layer Core of DSRC

4.27
LID
Logical Link Control Identifier

4.28
LLC
Logical Link Control

4.29
LPDU
LLC Protocol Data Unit

4.30
M_a
ManufacturerID

[EN 12834]

4.31
MAC
Medium Access Control

4.32
MMI
Man-Machine Interface

4.33
n.a.
not applicable

4.34
OBE
On-Board Equipment

4.35
OBU
On-Board Unit

4.36
ORSE
Organization for Road System Enhancement

4.37
$P_{a,b,c,d}$
Profile

[EN 13372, EN 12834]

EXAMPLE P_0 denotes Profile 0.

4.38
PDU
Protocol Data Unit

4.39
PER
Packed Encoding Rules

[ISO/IEC 8825-2]

4.40
PoC
Point of Control

4.41
PoO
Point of Observation

4.42
PPDU
Physical Layer Protocol Data Unit

4.43
PrWA
Private Window Allocation

[EN 12795]

4.44
PrWRq
Private Window Request

[EN 12795]

4.45
req
request

4.46
rs
response

4.47
RSE
Roadside Equipment

4.48
RTTT
Road Transport and Traffic Telematics

4.49
SAM
Secure Application Module

4.50
T-APDU
Transfer-Application Protocol Data Unit

4.51
T-ASDU
Transfer-Application Service Data Unit

4.52
T-Kernel
Transfer Kernel

4.53
TTI
Traffic and Traveller Information

4.54
VST
Vehicle Service Table

5 OBU and supporting information

The supplier shall provide the OBU, i.e. the DUT, and the associated information including:

- OBUs personalized to be able to perform tests according to the ICS and IXIT as defined in 5.1 and 5.2, respectively. At least five samples shall be submitted for test. More samples may be needed if several different data structures and data contents are required in the tests;
- user's manual for the OBU, which shall include instructions how to handle the equipment, and may include further detailed information about the protocol functions;
- implementation conformance statement according to 5.1. The ICS shall include statements regarding the following:
  - layer 7 services that are implemented in the OBU;
  - EFC functions (action types) that are implemented in the OBU;
  - whether or not data elements are used;
- implementation extra information for testing according to 5.2. The IXIT shall, if applicable, include:
  - a statement regarding which layer 2 services shall be used to transfer the L7 services (and EFC services);

— a description of security calculations in the OBU including a specification of the encryption algorithm used;

— values of the test Master Keys for calculation and verification of OBU security data such as authenticators and access credentials.

The supplier should also provide configuration/personalization equipment for the OBU if that ensures effective testing.

5.1 ICS

The ICS is a statement made by the supplier that claims conformance to a certain specification. The ICS states which capabilities have been implemented in the specifications. It also states possible limitations in the implementation of the specification.

This Technical Specification describes testing of implementations according to the following standards:

— DSRC application layer; EN 12834;

— EFC application interface; ISO 14906.

Annex A contains the ICS pro forma that shall be used for the ICS.

5.2 IXIT

The IXIT is a statement made by the supplier or an implementer of an IUT which contains or references all of the information, in addition to that given in the implementation conformance statement, related to the DUT and its testing environment. The IXIT enables the test laboratory to run an appropriate test suite against the DUT.

In this Technical Specification, the IXIT specifies the services in the lower communication layers that shall be used to perform the services to be tested. These services are described e.g. in EN 12795, EN 12253 and EN 13372.

The IXIT shall also contain further information and describe algorithms and procedures that are not specified in the above standards but are prerequisites to perform the testing. Example of such information is:

— content of the ApplicationContextMark in the VST;

— calculation of access credentials in DSRC services;

— use of ReturnCode in DSRC L7 services and EFC functions; and

— calculation of authenticators in EFC functions.

Annex B contains the IXIT pro forma that shall be used for the declaration of the IXIT.

6 Testing requirements

6.1 Conceptual test architecture

A remote test method shall be used for the conformance test of OBUs (DUT). Figure 4 shows the conceptual testing architecture of tester and DUT. The conformance test is only related to the implementation of the DUT.

The PoC, which is the point where the test events are controlled, shall be implemented inside the tester on the interface between the application layer and the application process.

The PoO, which is the point where the occurrence of test events is to be observed, shall be implemented inside the tester on the interface between the application layer and the application process. The PoO must also interface the data link layer since some tests require observation of the behaviour on this layer.

The PoC and PoO are characterized by a set of ASPs, according to the specifications of the tests. The test events observed at the PoO may be supplemented by information provided by other sub-units, e.g. MMI or an IC card if available with the DUT.

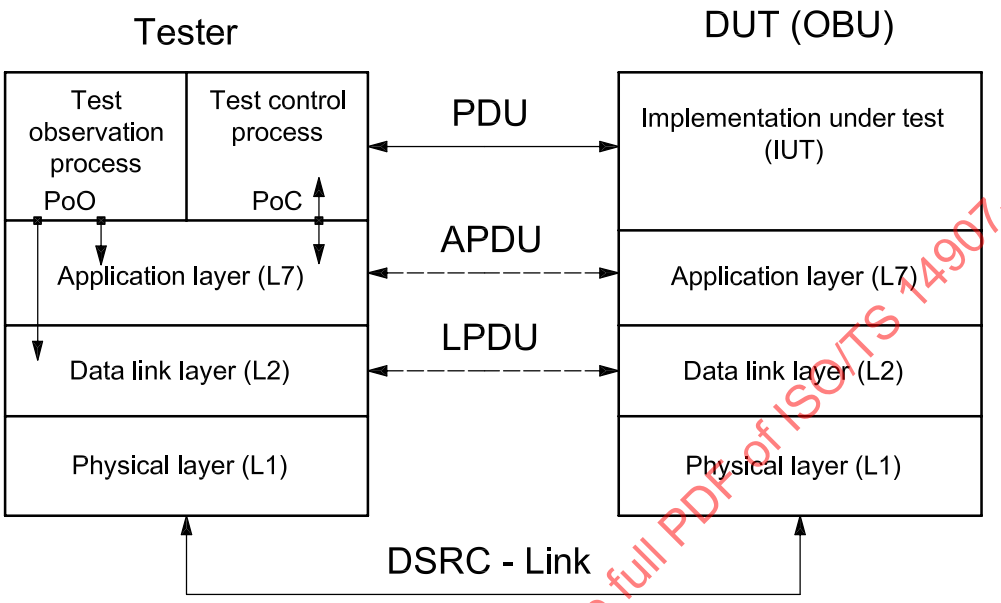

Figure 4 — Conceptual architecture of the conformance test

6.2 Conformance test system

The conformance tests shall be performed under controlled conditions by using a test system according to Figure 5.

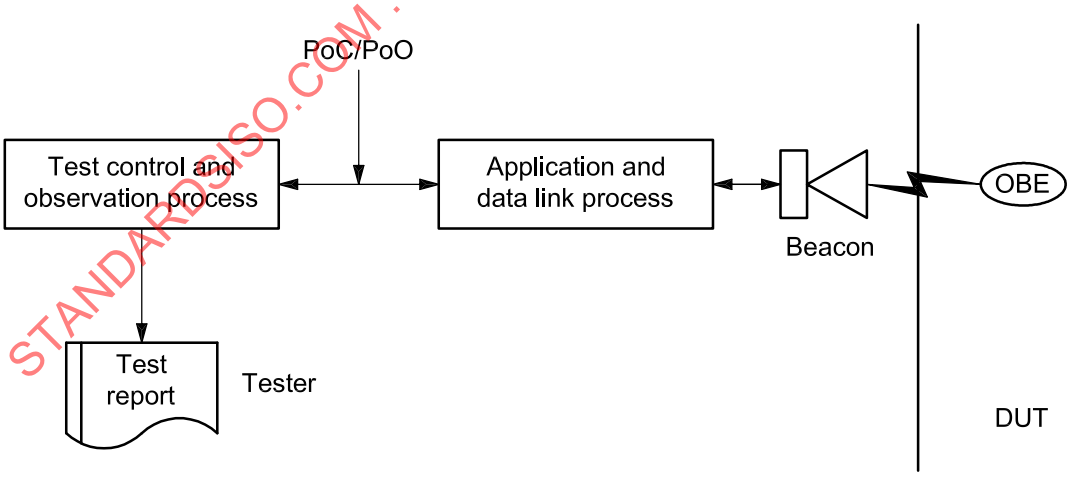

Figure 5 — Test facilities

6.2.1 Functionality of tester

6.2.1.1 Reference beacon

The beacon shall be a transparent physical layer interface for DSRC according to European or International Standards, which means it shall be able to operate standardized data profiles of DSRC links. A simple beacon may be used which will only send and receive frames to or from the DUT. The communication is controlled by the application and data link process of the tester.

Beacon parameters, e.g. the transmitting power, shall be adjustable by the tester.

6.2.1.2 Application and data link process

The tester shall be able to operate the communication link in real time according to data link layer functionality of European or International Standards. The process shall provide the connection to the test control and observation process and shall control the sending and reception of frames to and from the beacon.

6.2.1.3 Test control and observation processes

The test control and observation processes shall be implemented on an appropriately equipped computer system. The processes shall comply with the definitions of the application and data link layers defined by European or International standards and shall be able to generate test suites in reference to ISO 14906. The control process shall provide functionality to configure the test-bed, to generate test suites and to perform test suites.

EXAMPLE Test suites may be generated in plain ASCII format and may be retrieved by the real-time control process via FTP.

While the tests are being performed, all events shall be observed by the PoO process. All results shall be documented, visualized and be made available for detailed analysis.

6.2.2 Conformance testing

The tests performed by the tester shall be able to concentrate on EFC-communication protocol aspects, which are:

— initialization phase (BST-VST exchange);

— application layer services according to EN 12834;

— EFC functions and services according to ISO 14906;

— EFC data and parameters used according to ISO 14906; and

— transaction phase (logical sequence).

The DUT shall be operated in a controlled environment according to the manufacturer's instructions.

All test suites shall be generated corresponding to the specification of the manufacturer, which shall describe the behaviour of the DUT in terms of expected messages from the beacon and intended answers from the DUT. The communication sequence shall be unambiguously defined. The test generation tool shall be able to check the messages against the European or International DSRC standards. All events of a DUT are stored by the tester.

The test suites generated shall strictly correspond to the European or International DSRC standards and to ISO 14906.

The results of the tests shall only be related to the IUT of the DUT. Only the correctness of the implementation shall be tested, and any assessment of performance, robustness or reliability is excluded. The results of the tests shall be able to provide confidence that the DUT works according to ICS, the underlying DSRC standards and to ISO 14906, within the scope of the performed tests.

6.3 Test documentation

The test documentation shall provide comprehensive information of the used tester, the DUT, the tests carried out, and the gathered test results.

6.3.1 Tester

All relevant information and specifications of the used tester shall be documented in the test report.

The ICS and the IXIT shall be provided by the manufacturer or supplier of the DUT. The ICS and IXIT shall describe all the capabilities and limitations of the DUT according to Annex A and Annex B, respectively.

6.3.2 Test methods and test cases

A test suite is required which covers all implemented and specified functions of the DUT to be tested based on the provided ICS and IXIT. Completeness and correctness of the test suite is required to verify conformity between IUT and specification, including the referenced standards.

The test suite is separated in test groups; each test group consists of a number of test cases. The tester shall provide a tool which enables the user to specify the test suite and which provides documentation of the test suite.

The user-defined test suite should be automatically translated in a computer form which is executable by the tester.

6.3.3 Test results

Each test case carried out shall be documented by the test generation equipment. For each test case, a verdict shall be given: passed, inconclusive or failed. The results of all test cases shall form the conformance test report.

Annex A
(normative)

Implementation conformance statement pro forma

The forms in this annex shall be completed by the supplier of the DUT (OBU) to state conformance to applicable parts in EN 12834 (DSRC L7 standard) and ISO 14906.

When the forms are completed, this annex will form the ICS that shall be submitted to the test laboratory before start of the tests.

A.1 Identification of OBU supplier

Table A.1 — Identification of OBU supplier form

Company	
Postal address	
Telephone	
Contact person	
E-mail address	

A.2 Identification of OBU

Table A.2 — Identification of OBU form

Brand	
Type, Version	
ManufacturerID	
EquipmentClass	
Serial numbers of supplied units	

A.3 DSRC application layer implementation

A.3.1 Declaration of application layer features

The EN 12834 (DSRC L7) implemented functionality shall be declared by completing Table A.3.

Table A.3 — EN 12834 functionality

Feature(s)			Status	Implemented Yes/No/n.a.	Restrictions or other comments
T-KE	Fragmentation/defragmentation		optional/mand.		
	Concatenation/deconcatenation		optional/mand.		
	Multiplexing/demultiplexing		optional/mand.		
	Fragmentation header	1 octet	mand.		
		2 octets	optional/mand.		
		3 octets	optional/mand.		
	Service Primitives	GET	optional		
		SET	optional		
		ACTION	optional		
		EVENT-REPORT	mand.		
		INITIALIZATION	mand.		
I-KE			optional/mand.		
Timer T (second)			255/0-255		
LID for INITIALIZATION.req			broadcast/ private		

NOTE "/" means selecting any number or feature; "n.a." means "not applicable".

A.3.2 Detailed L7 services statement

For each implemented L7 service, the corresponding table below shall be completed. The information is used to control the tests of the service.

A.3.2.1 INITIALIZATION

Table A.4 — INITIALIZATION.req, BST

Data Element	Data Type	Used/Not used, Restrictions
rsu	BeaconID	
time	Time	
profile	Profile	
mandApplications	ApplicationList	
nonmandApplications	ApplicationList, OPTIONAL	
profileList	SEQUENCE (SIZE (0..127,...)) OF Profile	

Table A.5 — INITIALIZATION.rs, VST

Data Element	Data Type	Used/Not used, Restrictions
profile	Profile	
applications	ApplicationList	
obeConfiguration	ObeConfiguration	

A.3.2.2 GET

Table A.6 — GET.req

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
accessCredentials	OCTET STRING OPTIONAL	
iid	DsrcEID OPTIONAL	
attrIdList	AttributeIdList OPTIONAL	

If access credentials are used, the algorithm for how to calculate these shall be described in the IXIT.

Table A.7 — GET.rs

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
attributelist	AttributeList OPTIONAL	
ret	ReturnStatus OPTIONAL	

A.3.2.3 SET

Table A.8 — SET.req

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN	
eid	Dsrc-EID	
accessCredentials	OCTET STRING OPTIONAL	
attrList	AttributeList	
iid	DsrcEID OPTIONAL	

If access credentials are used, the algorithm for how to calculate these shall be described in the IXIT.

Table A.9 — SET.rs

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
ret	ReturnStatus OPTIONAL	

A.3.2.4 EVENT-REPORT

Table A.10 — EVENT-REPORT.req

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN	
eid	Dsrc-EID	
eventType	EventType	
accessCredentials	OCTET STRING (SIZE (0..127,..)) OPTIONAL	
eventParameter	Container OPTIONAL	
iid	DsrcEID OPTIONAL	

If access credentials are used, the algorithm for how to calculate these shall be described in the IXIT.

Table A.11 — EVENT-REPORT.rs

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
ret	ReturnStatus OPTIONAL	

A.4 EFC Functions

A.4.1 Declaration of EFC functions implemented

The ISO 14906-implemented EFC functions (DSRC L7 ACTION-services) shall be declared by completing Table A.12.

Table A.12 — EFC Functions

EFC function	Implemented Yes/No	Restrictions and remarks
GET_STAMPED		
SET_STAMPED		
GET_SECURE		
SET_SECURE		
GET_INSTANCE		
SET_INSTANCE		
GET_NONCE		
SET_NONCE		
TRANSFER_CHANNEL		
COPY		
SET_MMI		
SUBTRACT		
ADD		
DEBIT		
CREDIT		
ECHO		

A.4.2 Detailed EFC functions statement

For each implemented EFC function, the corresponding table below shall be completed. The information is used to control the tests of the function.

A.4.2.1 GET STAMPED

Table A.13 — ACTION.rq (GET STAMPED)

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN = TRUE	
eid	Dsrc-EID	
actionType	ActionType = 0	
accessCredentials	OCTET STRING OPTIONAL	
actionParameter	GetStampedRq ::= SEQUENCE { attributeIdList AttributeIdList, nonce OCTET STRING, keyRef INTEGER(0..255) }	
iid	DsrcEID OPTIONAL	

If access credentials are used, the algorithm for how to calculate these shall be described in the IXIT.

Table A.14 — ACTION.rs (GET STAMPED)

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
responseParameter	GetStampedRs::= SEQUENCE { attributeList AttributeList, authenticator OCTET STRING }	
ret	ReturnStatus OPTIONAL	

The algorithm for calculation of the authenticator shall be described in the IXIT.

A.4.2.2 SET STAMPED

Table A.15 — ACTION.req (SET STAMPED)

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN = TRUE	
eid	Dsrc-EID	
actionType	ActionType = 1	
accessCredentials	OCTET STRING OPTIONAL	
actionParameter	SetStampedRq::= SEQUENCE { attributeList AttributeList, nonce OCTET STRING, keyRef INTEGER(0..255) }	
iid	DsrcEID OPTIONAL	

If access credentials are used, the algorithm for how to calculate these shall be described in the IXIT.

Table A.16 — ACTION.rs (SET STAMPED)

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
responseParameter	OCTET STRING	
ret	ReturnStatus OPTIONAL	

The algorithm for calculation of the authenticator in the responseParameter OCTET STRING shall be described in the IXIT.

A.4.2.3 GET SECURE

Table A.17 — ACTION.req (GET SECURE)

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN = TRUE	
eid	Dsrc-EID	
actionType	ActionType = 2	
accessCredentials	OCTET STRING OPTIONAL	
actionParameter	OCTET STRING	
iid	DsrcEID OPTIONAL	

If access credentials are used, the algorithm for how to calculate these shall be described in the IXIT.

Table A.18 — ACTION.rs (GET SECURE)

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
responseParameter	OCTET STRING	
ret	ReturnStatus OPTIONAL	

The content in the responseParameter OCTET STRING shall be described in the IXIT.

A.4.2.4 SET SECURE

Table A.19 — ACTION.req (SET SECURE)

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN	
eid	Dsrc-EID	
actionType	ActionType = 3	
accessCredentials	OCTET STRING OPTIONAL	
actionParameter	OCTET STRING	
iid	DsrcEID OPTIONAL	

If access credentials are used, the algorithm for how to calculate these shall be described in the IXIT.

The content in the actionParameter OCTET STRING shall be described in the IXIT.

Table A.20 — ACTION.rs (SET SECURE)

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
responseParameter	OCTET STRING OPTIONAL	
ret	ReturnStatus OPTIONAL	

The content in the responseParameter OCTET STRING shall be described.

A.4.2.5 GET INSTANCE

Table A.21 — ACTION.req (GET INSTANCE)

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN = TRUE	
eid	Dsrc-EID	
actionType	ActionType = 4	
accessCredentials	OCTET STRING OPTIONAL	
actionParameter	GetInstanceRq ::= SEQUENCE { posOfFirstInstance INTEGER(0..255), posOfLastInstance INTEGER(0..255), attributeIdList AttributeIdList}	
iid	DsrcEID OPTIONAL	

If access credentials are used, the algorithm for how to calculate these shall be described in the IXIT.

Table A.22 — ACTION.rs (GET INSTANCE)

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
responseParameter	GetInstanceRs::= SEQUENCE (0..127....) OF SEQUENCE { attributeId INTEGER(0..127....), attributeValues Container::=OCTET STRING }	
ret	ReturnStatus OPTIONAL	

A.4.2.6 SET INSTANCE

Table A.23 — ACTION.req (SET INSTANCE)

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN	
eid	Dsrc-EID	
actionType	ActionType = 5	
accessCredentials	OCTET STRING OPTIONAL	
actionParameter	SetInstanceRq::= SEQUENCE { posOfInstance INTEGER(0..255), attribute Attributes}	
iid	DsrcEID OPTIONAL	

If access credentials are used, the algorithm for how to calculate these shall be described in the IXIT.

Table A.24 — ACTION.rs (SET INSTANCE)

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
responseParameter	NONE	
ret	ReturnStatus OPTIONAL	

A.4.2.7 GET NONCE

Table A.25 — ACTION.req (GET NONCE)

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN = TRUE	
eid	Dsrc-EID	
actionType	ActionType = 6	
accessCredentials	OCTET STRING n.a.	
actionParameter	NONE	
iid	DsrcEID OPTIONAL	

Table A.26 — ACTION.rs (GET NONCE)

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
responseParameter	OCTET STRING	
ret	ReturnStatus OPTIONAL	

A.4.2.8 SET NONCE

Table A.27 — ACTION.req (SET NONCE)

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN	
eid	Dsrc-EID	
actionType	ActionType = 7	
accessCredentials	OCTET STRING n.a.	
actionParameter	OCTET STRING	
iid	DsrcEID OPTIONAL	

Table A.28 — ACTION.rs (SET NONCE)

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
responseParameter	NONE	
ret	ReturnStatus OPTIONAL	

A.4.2.9 TRANSFER CHANNEL

Table A.29 — ACTION.req (TRANSFER CHANNEL)

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN	
eid	Dsrc-EID	
actionType	ActionType = 8	
accessCredentials	NONE	
actionParameter	ChannelRq ::= SEQUENCE { channelId ChannelID, apdu OCTET STRING }	
iid	DsrcEID OPTIONAL	

Table A.30 — ACTION.rs (TRANSFER CHANNEL)

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
responseParameter	ChannelRs ::= SEQUENCE { channelId ChannelID, apdu OCTET STRING }	
ret	ReturnStatus OPTIONAL	

A.4.2.10 COPY

Table A.31 — ACTION.req (COPY)

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN	
eid	Dsrc-EID	
actionType	ActionType = 9	
accessCredentials	OCTET STRING OPTIONAL	
actionParameter	CopyRq ::= SEQUENCE { destinationEID INTEGER(0..127....), attributeIdList AttributeIdList, }	
iid	DsrcEID OPTIONAL	

If access credentials are used, the algorithm for how to calculate these shall be described in the IXIT.

Table A.32 — ACTION.rs (COPY)

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
responseParameter	NONE	
ret	ReturnStatus OPTIONAL	

A.4.2.11 SET MMI

Table A.33 — ACTION.req (SET MMI)

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN	
eid	Dsrc-EID	
actionType	ActionType = 10	
accessCredentials	NONE	
actionParameter	SetMMIRq ::= INTEGER { ok (0) nok (1) contactOperator (2) reservedForFutureCENUse (3..127), reservedForPrivateUse (128..255) } (0..255)	
iid	DsrcEID OPTIONAL	

Table A.34 — ACTION.rs (SET MMI)

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
responseParameter	NONE	
ret	ReturnStatus OPTIONAL	

A.4.2.12 SUBTRACT

Table A.35 — ACTION.req (SUBTRACT)

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN	
eid	Dsrc-EID	
actionType	ActionType = 11	
accessCredentials	OCTET STRING OPTIONAL	
actionParameter	SubRq::= SEQUENCE { attributeId INTEGER(0..127....), value INTEGER }	
iid	DsrcEID OPTIONAL	

If access credentials are used, the algorithm for how to calculate these shall be described in the IXIT.

Table A.36 — ACTION.rs (SUBTRACT)

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
responseParameter	NONE	
ret	ReturnStatus OPTIONAL	

A.4.2.13 ADD

Table A.37 — ACTION.req (ADD)

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN	
eid	Dsrc-EID	
actionType	ActionType = 12	
accessCredentials	OCTET STRING OPTIONAL	
actionParameter	AddRq::= SEQUENCE { attributeId INTEGER(0..127....), value INTEGER }	
iid	DsrcEID OPTIONAL	

If access credentials are used, the algorithm for how to calculate these shall be described in the IXIT.

Table A.38 — ACTION.rs (ADD)

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
responseParameter	NONE	
ret	ReturnStatus OPTIONAL	

A.4.2.14 DEBIT

Table A.39 — ACTION.req (DEBIT)

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN = TRUE	
eid	Dsrc-EID	
actionType	ActionType = 13	
accessCredentials	OCTET STRING OPTIONAL	
actionParameter	DebitRq ::= SEQUENCE { debitPaymentFee PaymentFee, nonce OCTET STRING, keyRef INTEGER(0..255) }	
iid	DsrcEID OPTIONAL	

If access credentials are used, the algorithm for how to calculate these shall be described in the IXIT.

Table A.40 — ACTION.rs (DEBIT)

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
responseParameter	DebitRs::= SEQUENCE { debitResult ResultFin, debitAuthenticator OCTET STRING }	
ret	ReturnStatus OPTIONAL	

The algorithm for calculation of the authenticator shall be described in the IXIT.

A.4.2.15 CREDIT

Table A.41 — ACTION.req (CREDIT)

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN = TRUE	
eid	Dsrc-EID	
actionType	ActionType = 14	
accessCredentials	OCTET STRING OPTIONAL	
actionParameter	CreditRq ::= SEQUENCE { refund PaymentFee, nonce OCTET STRING, keyRef INTEGER(0..255) }	
iid	DsrcEID OPTIONAL	

If access credentials are used, the algorithm for how to calculate these shall be described in the IXIT.

Table A.42 — ACTION.rs (CREDIT)

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
responseParameter	CreditRs::= SEQUENCE { creditResult ResultFin, creditAuthenticator OCTET STRING }	
ret	ReturnStatus OPTIONAL	

The algorithm for calculation of the authenticator shall be described in the IXIT.

A.4.2.16 ECHO

Table A.43 — ACTION.req (ECHO)

Data Element	Data Type	Used/Not used, Restrictions
mode	BOOLEAN	
eid	Dsrc-EID	
actionType	ActionType = 15	
accessCredentials	NONE	
actionParameter	OCTET STRING	
iid	DsrcEID OPTIONAL	

Table A.44 — ACTION.rs (ECHO)

Data Element	Data Type	Used/Not used, Restrictions
eid	Dsrc-EID	
iid	DsrcEID OPTIONAL	
responseParameter	OCTET STRING	
ret	ReturnStatus OPTIONAL	

Annex B
(normative)

Implementation of extra information for testing

The forms in this annex shall be completed by the supplier of the DUT (i.e. OBU) to define extended functionality and parameter settings that are necessary for performing the tests.

When the forms are completed, this annex will form the IXIT that shall be submitted to the test laboratory before start of the tests.

B.1 Identification of OBU supplier

Table B.1 — Identification of OBU supplier

Company	
Postal address	
Telephone	
Contact person	
E-mail address	

B.2 Identification of OBU

Table B.2 — Identification of OBU

Brand	
Type, version	
ManufacturerID	
EquipmentClass	
Serial numbers on supplied units	

B.3 DSRC L1 Implementation

B.3.1 DSRC L1 standards / specifications

Table B.3 — DSRC L1 standards / specifications

Standard/Specification	Restrictions and remarks

B.3.2 DSRC L1 parameter settings

Table B.4 — DSRC L1 parameter settings

Parameter	Setting

B.4 DSRC L2 Implementation

B.4.1 DSRC L2 standards/specifications

Table B.5 — DSRC L2 standards / specifications

Standard/specification	Restrictions and remarks

B.4.2 DSRC L2 services implemented for transferring of L7 APDUs

Table B.6 — DSRC L2 services implemented for transferring of L7 APDUs

L7 APDU	DSRC L2 service implemented
INITIALIZATION.req	
INITIALIZATION.rs	
GET.req	
GET.rs	
SET.req	
SET.rs	
ACTION.req	
ACTION.rs	
EVENT_REPORT.req	
EVENT_REPORT.rs	

B.5 Security Implementation

B.5.1 Security standards/specifications

Table B.7 — Security standards/specifications

Standard/specification	Restrictions and remarks

B.5.2 Calculation of access credentials

Table B.8 — Calculation of access credentials in different L7 services

L7 service	Algorithm for calculation of access credentials
GET.req	
SET.req	
ACTION.req	
EVENT_REPORT.req	

B.5.3 Calculation of authenticators

Table B.9 — Calculation of the authenticator in different EFC functions

L7 service	Algorithms for calculation of the authenticator
GET_STAMPED	
SET_STAMPED	
GET_SECURE	
SET_SECURE	
DEBIT	
CREDIT	

B.5.4 Test master key values

NOTE More than one keyRef may be applicable for each function.

Table B.10 — Test master keys

L7 service that uses access credentials	keyRef	Test master key value
GET.request	n.a.	
SET.request	n.a.	
ACTION.request	n.a.	
EVENT_REPORT.request	n.a.	
EFC function that uses authenticator		
GET_STAMPED		
SET_STAMPED		
GET_SECURE		
SET_SECURE		
DEBIT		
CREDIT		

B.6 Return Code

Table B.11 — Return code

Value	Description	Usage
0	noError: The requested operation was performed successfully.	
1	accessDenied: The requested operation was not performed for reasons pertinent to the security system.	
2	argumentError: One or more attribute values were not accessed because the identifier for the specified attribute was not recognized or the attribute value specified was out of range or otherwise inappropriate for one or more attributes, or the action or event-report invoked was not supported by the receiving entity.	
3	complexityLimitation: The requested operation was not performed because a parameter was too complex.	
4	processingFailure: A general failure in processing the operation was encountered.	
5	processing: The requested operation is being processed, and the result is not yet available.	
6	chainingError: The requested operation was not performed in accordance with the rule defined in 6.3.8 (on concatenation with chaining) in EN 12834.	

Annex C
(informative)

OBU test cases

In this annex, examples of test cases for DSRC and EFC functionality are presented. The annex does not intend to specify a complete test suite for a certain implementation. To compose a test suite for a specific EFC implementation the test cases may have to be modified and new test cases may have to be defined and added in order for the conformance test to be complete.

Tables C.1-C.4 below provide an overview of the OBU test cases:

— Table C.1 contains a test suite for testing the basic application layer features as defined in EN 12834 (Layer 7 standard);

— Table C.2 contains a test suite for testing the basic EFC functions as defined in ISO 14906;

— Table C.3 contains a test suite for testing an EFC transaction, on an application and application interface level;

— Table C.4 contains a test suite for testing extended DSRC-EFC features as defined in EN 12795 (Layer 2 standard).

Each test suite is specified in terms of test group, test case ID, test case name and remarks.

Table C.1 — Overview of the basic application layer test suite

Test Group	Test Case ID	Test Case Name	Remarks
DSRC L7 INITIALIZATION	TC01-A	Profile handling	
	TC01-B	Application ID handling	
	TC01-C	Time handling	
	TC01-D	Beacon ID handling	
DSRC L7 EVENT_REPORT RELEASE	TC02-A	OBU Register/De-register application	
DSRC L7 GET	TC03-A		
DSRC L7 SET	TC04-A		

Table C.2 — Overview of the basic EFC functions test suite

Test Group	Test Case ID	Test Case Name	Remarks
EFC GET_STAMPED	TC10-A		
EFC GET-SECURE	TC12-A		
EFC SET-SECURE	TC13-A		
EFC SET_MMI	TC20-A		
EFC ECHO	TC25-A		

NOTE The numbering of the test cases in Table C.2 has been made considering future extensions to include test cases for all EFC functions defined in ISO 14906.

Table C.3 — Overview of an EFC transaction test suite

Test Group	Test Case ID	Test Case Name	Remarks
EFC transaction functionality	TC30-A	CARDME transaction	As defined in Annex B of ISO 14906

Table C.4 — Overview of the extended DSRC EFC test suite

Test Group	Test Case ID	Test Case Name	Remarks
DSRC INITIALIZATION Recovery	TC40.A	Handling of INITIALIZATION. PrWRq, VST lost.	
DSRC ACn Recovery	TC41-A	Fast Access Recovery Functions: GET/SET-rs lost	
	TC41-B	Slow Access Recovery Functions: GET/SET-rs lost	
DSRC Communication Blocked – Recovery	TC42-A	Communication Blocked – Recovery	
	TC42-B	Communication Blocked (300 s) – New connection	

The test cases (TC) associated with Tables C.1-C.4 are defined in the subsequent clauses.

Every test shall be repeated 10 times, unless otherwise explicitly specified.

C.1 Basic application layer test suite

C.1.1 TC01 DSRC (L7) INITIALIZATION

C.1.1.1 TC01-A profile handling

C.1.1.1.1 Equipment and set-up

The test set-up shall be according to what is described in Clause 6.

OBUs according to Clause 5 shall be used. The supplier's ICS and IXIT shall include information concerning:

— layer 2 services that are used for transmission of the layer 7 APDUs;

— the supported application(s); and

— the supported profile(s).

In this test case, it is assumed that the OBU:

— contains at least an EFC application;

— supports profiles P_a and P_b; and

— does not support profiles P_c and P_d.

M_a is a manufacturerId defined in the test system.

The equipment shall be placed in such a way that optimal communication can be foreseen.

C.1.1.1.2 Main execution steps

Main execution steps comprise tests concerning:

— steps 1-2: Handling of profile P_a;

— steps 3-4: Handling of profile P_b;

— steps 5-6: Handling of multiple profiles P_a, P_c;

— steps 7-8: Handling of multiple profiles P_b, P_d;

— steps 9-10: Handling of multiple profiles P_c, P_d, P_a; and

— steps 11-12: Handling of profile P_d.

Table C.5 — Execution steps for TC01-A profile handling

Step	Action	Expected response	Note
1	Configure the BST according to Annex A in ISO 14906 with: mandatory application = EFC profile = P_a beaconId manufacturerid = M_a IndividualId = 1		It is assumed that the OBU has not received any BST with this beaconId during the last 255 s
2	Send BST every 5-10 ms until an initialization is performed and a VST is received	VST according to Annex A in ISO 14906, profile = P_a	Profile P_a expected from OBU
3	Same as step 1 except for: profile = P_b		
4	Same as step 2	VST according to Annex A in ISO 14906, profile = P_b	Profile P_b expected from OBU
5	Same as step 1 except: profile = P_a profileList = P_c		
6	Same as step 2	VST according to Annex A in ISO 14906, profile = P_a	Profile P_a expected from OBU
7	Same as step 1 except: profile = P_b profileList = P_d		
8	Same as step 2	VST according to Annex A in ISO 14906, profile = P_b	Profile P_b expected from OBU
9	Same as step 1 except: profile = P_c profileList = P_d, P_a		
10	Same as step 2.	VST according to Annex A in ISO 14906, profile = P_a	Profile P_a expected from OBU
11	Same as step 1 except: profile = P_c		
12	Same as step 2	None	Profile P_c not supported by the OBU

C.1.1.1.3 Acceptance criteria

Table C.6 — TC01-A acceptance criteria

Item	Acceptance criteria	Limit	Note
1	INITIALIZATION.rs,VST depending on profile(s) in BST	VST with profile P_a in steps 2, 6, 10 VST with profile P_b in steps 4 and 8 No VST in step 12	

C.1.1.2 TC01-B application Id handling

C.1.1.2.1 Equipment and set-up

The test set-up shall be according to what is described in Clause 6.

OBUs according to Clause 5 shall be used. The supplier's ICS and IXIT shall include information concerning:

— layer 2 services that are used for transmission of the layer 7 APDUs;

— the application(s) supported; and

— the profile(s) supported.

In this test case, it is assumed that the OBU:

— contains an EFC application;

— does not contain any TTI or AVI application; and

— supports profile P_a.

M_a is a manufacturerId defined in the test system.

The equipment shall be placed in such a way that optimal communication can be foreseen.

NOTE The OBU is expected to support at least the EFC application. Furthermore, the OBU may be configured with a single application (EFC) or multiple applications.

C.1.1.2.2 Main execution steps

The main execution steps comprise tests concerning:

— steps 1-2: Handling of mandatory application (= TTI);

— steps 3-4: Handling of mandatory applications (= EFC, TTI);

— steps 5-6: Handling of mandatory applications (= TTI, EFC);

— steps 7-8: Handling of mandatory applications (= EFC) and non-mandatory applications (TTI and AVI); and

— steps 9-10: Handling of mandatory applications (= TTI) and non-mandatory applications (EFC and AVI).

Table C.7 — Execution steps for TC01-B application id handling

Step	Action	Expected response	Note
1	Configure the BST according to Annex A in ISO 14906 with: profile = P_a mandatory applications = TTI beaconId manufacturerID = M_a IndividualId = 1		It is assumed that the OBU has not received any BST with this beaconId during the last 255 s
2	Send BST every 5-10 ms (until initialization is performed and a VST is received)	None	The TTI application is not supported by the OBU
3	Same as step 1 except that mandatory applications = EFC, TTI		
4	Same as step 2	VST according to Annex A in ISO 14906 with: applications = EFC	
5	Same as step 1 except that mandatory applications = TTI, EFC		
6	Same as step 2	VST according to Annex A in ISO 14906 with: applications = EFC	
7	Same as step 1 except that mandatory applications = EFC non-mandatory applications = TTI, AVI		
8	Same as step 2	VST according to Annex A in ISO 14906 with: applications = EFC	
9	Same as step 1 except that mandatory applications = TTI non-mandatory applications = EFC, AVI		
10	Same as step 2	VST according to Annex A in ISO 14906 with: applications = EFC	

C.1.1.2.3 Acceptance criteria

Table C.8 — TC01-B acceptance criteria

Item	Acceptance criteria	Limit	Note
1	INITIALIZATION.rs ,VST depending on application(s) in BST	No VST in step 2 VST with application EFC in step 4, step 6, step 8, and step 10	

C.1.1.3 TC01-C time handling

C.1.1.3.1 Equipment and set-up

The test set-up shall be according to what is described in Clause 6.

OBUs according to Clause 5 shall be used. The supplier's ICS and IXIT shall include information concerning:

— layer 2 services that are used for transmission of the layer 7 APDUs;

— the application(s) supported; and

— the profile(s) supported.

In this test case, it is assumed that the OBU:

— contains an EFC application;

— supports profile P_a.

M_a is a manufacturerId defined in the test system.

The time in the BST shall be UNIX-time. In the test, the time is instantly increased to a certain value to test the OBU behaviour. This corresponds to a certain elapsed time and will shorten the time for execution of the test.

The equipment shall be placed in such a way that optimal communication can be foreseen.

C.1.1.3.2 Main execution steps

Main execution steps comprise:

— steps 1-4: Handling of time in BST = current UNIX-time;

— steps 5-7: Handling of BST time set to current UNIX-time + 255 seconds; and

— steps 8-10: Handling of BST for 100 seconds, no BST for 100 seconds, BST for 100 seconds.

Table C.9 — Execution steps for TC01-C time handling

Step	Action	Expected response	Note
1	Configure the BST according to Annex A in ISO 14906 with: profile = P_a mandatory applications = EFC beaconId manufacturerid = M_a IndividualId = 1 Time in BST = UNIX-time		It is assumed that the OBU has not received any BST with this beaconId during the last 255 s
2	Send BST every 5-10 ms (until initialization is performed and a VST is received)	VST according to Annex A in ISO 14906	t-ref is the time in the BST that was answered by the VST
3	Send a GET/SET/ACTION request in confirmed mode (mode = TRUE); select the L7 service according to the capabilities of the OBU	Response to the request	
4	Send BST every 5-10 ms until 3 BSTs have been sent	None	No answer shall be received from OBU
5	Configure the BST according to step 1 Time in BST = t-ref + 255		
6	Send BST every 5-10 ms (until initialization is performed and a VST is received)	VST according to Annex A in ISO 14906	Time in BST is now t-ref + 255 s; the OBU is expected to respond since Time – t-ref > 255 s
7	Send a GET/SET/ACTION request in confirmed mode (mode = TRUE); select the L7 service according to the capabilities of the OBU	Response to the request	
8	Send BST every 5-10 ms for 100 s	None	
9	Send no BST for 100 s	None	
10	Send BST every 5-10 ms for 100 s	None	

C.1.1.3.3 Acceptance criteria

Table C.10 — TC01-C acceptance criteria

Item	Acceptance criteria	Limit	Note
1	INITIALIZATION.rs ,VST depending on time in BST	VST is sent in steps 2 and 6 No VST is sent in steps 4, 8 and 10	

C.1.1.4 TC01-D beacon ID handling

C.1.1.4.1 Equipment and set-up

The test set-up shall be according to what is described in Clause 6.

OBUs according to Clause 5 shall be used. The supplier's ICS and IXIT shall include information concerning:

— layer 2 services that are used for transmission of the layer 7 APDUs;

— the application(s) supported;

— the profile(s) supported.

In this test case, it is assumed that the OBU:

— contains an EFC application; and

— supports profile P_a.

M_a and M_b are different values of manufacturerId defined in the test system.

The equipment shall be placed in such a way that optimal communication can be foreseen.

C.1.1.4.2 Main execution steps

Main execution steps comprise changing of Beacon ID according to the steps below:

steps 1-4: Handling of manufacturerID (unchanged) and individual id (unchanged);

steps 5-8: Handling of manufacturerID (changed) and individual id (unchanged); and

steps 9-10: Handling of manufacturerID (unchanged) and individual id (changed).

Table C.11 — Execution steps for TC01-D beacon id handling

Step	Action	Expected response	Note
1	Configure the BST according to Annex A in ISO 14906 with: profile = P_a mandatory applications = EFC beaconId manufacturerid = M_a IndividualId = 1		It is assumed that the OBU has not received any BST with this beaconId during the last 255 s
2	Send BST every 5-10 ms (until initialization is performed and a VST is received)	VST according to Annex A in ISO 14906	
3	Send a GET/SET/ACTION request in confirmed mode (mode = TRUE); select the L7 service according to the capabilities of the OBU	Response to the request	
4	Send BST every 5-10 ms until 3 BSTs have been sent	None	No answer shall be received from OBU
5	Same as step 1 except that beaconId manufacturerid = M_b		
6	Send the BST according to step 1; send BST every (5-10) ms until an uplink frame is received	VST according to Annex A in ISO 14906	It is assumed that this step is performed within the time 255 s from step 1 OBU responds to changed beaconId
7	Send a GET/SET/ACTION request in confirmed mode (mode = TRUE); select the L7 service according to the capabilities of the OBU	Response to the request	
8	Send BST every 5-10 ms until 3 BSTs have been sent	None	No answer shall be received from OBU
9	Same as step 1 except that beaconId manufacturerid = M_b IndividualId = 2		
10	Send the BST according to step 1; send BST every 5-10 ms until an uplink frame is received	VST according to Annex A in ISO 14906	It is assumed that this step is performed within the time 255 s from step 6 OBU responds to changed beaconId

C.1.1.4.3 Acceptance criteria

Table C.12 — TC01-D acceptance criteria

Item	Acceptance criteria	Limit	Note
1	INITIALIZATION.rs, VST depending on BeaconId	VST is sent in steps 2, 6 and 10 No VST is sent in steps 4 and 8	If a retransmission occurs during the test, the test shall be restarted

C.1.2 TC02 DSRC L7 EVENT_REPORT RELEASE

C.1.2.1 TC02-A OBU register/de-register application

C.1.2.1.1 Equipment and set-up

The test set-up shall be according to what is described in Clause 6.

OBUs according to Clause 5 shall be used. The supplier's ICS and IXIT shall include information concerning:

— layer 2 services that are used for transmission of the layer 7 APDUs;

— the application(s) supported; and

— the profile(s) supported.

In this test case, it is assumed that the OBU:

— contains an EFC application; and

— supports profile P_a.

M_a is a manufacturerId defined in the test system.

The equipment shall be placed in such a way that optimal communication can be foreseen.

C.1.2.1.2 Main execution steps

Main execution steps comprise the handling of EN 12834 EVENT_REPORT Request.

Table C.13 — Execution steps for TC02-A OBU register/de-register application

Step	Action	Expected response	Note
1	Configure the BST according to Annex A in ISO 14906 with: profile = P_a mandatory applications = EFC beaconId manufacturerid = M_a IndividualId = 1		It is assumed that the OBU has not received any BST with this beaconId during the last 255 s
2	Send BST every 5-10 ms (until initialization is performed and a VST is received)	VST according to Annex A in ISO 14906	
3	Send a GET/SET/ACTION request in confirmed mode (mode = TRUE); select the L7 service according to the capabilities of the OBU	Response to the request	This step will test that the OBU has registered the application and that it will answer to a request
4	Send an EVENT_REPORT-(RELEASE). request: without accessCredentials, mode = FALSE, eventType = RELEASE according to Annex A in ISO 14906	None	Since the Release command is sent in non-confirmed mode (mode = FALSE) no response shall be sent from the OBU; the OBU shall de-register the application
5	Send the same request as in step 3	None	The OBU shall not answer to the request

C.1.2.1.3 Acceptance criteria

Table C.14 — TC02-A acceptance criteria

Item	Acceptance criteria	Limit	Note
1	Answer to the GET/SET/ACTION.request	Response in step 3 No response in step 5	If a retransmission occurs during the test, the test shall be restarted

C.1.3 TC03 DSRC L7 GET

C.1.3.1 TC03-A basic DSRC L7 GET

C.1.3.1.1 Equipment and set-up

The test set-up shall be according to what is described in Clause 6.

OBUs according to Clause 5 shall be used. The supplier's ICS and IXIT shall include information concerning:

— set of correct (n1) and incorrect (n2) value for Eid;

— set of correct and incorrect values for accessCredentials (if implemented) and attributeIdList in the GET request test frames;

— set of expected values and returnCode in the GET response test frames related to above possible correct/incorrect values for Eid, accessCredentials and attributeIdList;

— algorithms and master keys values related to secure services inherent in the protocol (if implemented);

— layer 2 services that are used for transmission of the layer 7 APDUs.

The equipment shall be placed in such a way that optimal communication can be foreseen.

C.1.3.1.2 Main execution steps

Main execution steps include:

— step 1: Initialization

— steps 2-3: Eid; correct value, AccessCredentials; correct value, AttributeIdList; correct values;

— steps 4-5: Eid; correct value, AccessCredentials; incorrect value, AttributeIdList, correct values;

— steps 6-7: Eid; correct value, AccessCredentials; correct value, AttributeIdList; non-existing attribute; and

— steps 8-9: Eid; non-existing element, AccessCredentials; correct value, AttributeIdList; correct value.

Table C.15 — Execution steps for TC03-A basic DSRC L7 GET

Step	Action	Expected response	Note
1	Perform initialization according to TC-01 steps 1-2		
2	Configure a GET.request in confirmed mode (mode = True eid = n1 accessCredentials = OCTET STRING attributeIdList)		 Mode is always true ICS shall indicate a correct value for eid If present, ICS/IXIT shall indicate how to present a correct value for accessCredentials ICS shall indicate correct values for attributeIds
3	Send GET.request	Receive GET.response containing;(eid = n1 attributeList returnCode OPTIONAL)	 Same as in GET.request Requested attributes ICS/IXIT shall indicate whether returnCode is present
4	Configure a GET.request in confirmed mode (mode = True eid = n1 accessCredentials = OCTET STRING attributeIdList)		 Mode is always true ICS shall indicate a correct value for eid ICS/IXIT shall indicate how to present an incorrect value for accessCredentials ICS shall indicate correct values for attributeIds

Table C.15 (*continued*)

Step	Action	Expected response	Note
5	Send GET.request	Receive GET.response containing (GET.response eid = n1 attributeList returnCode OPTIONAL)	 Same as in GET.request Requested attributes not present due to incorrect accessCredentials ICS/IXIT shall indicate whether returnCode is present and the value
6	Configure a GET.request in confirmed mode (mode = True eid = n1 accessCredentials = OCTET STRING attributeIdList)		 Mode is always true ICS shall indicate a correct value for eid ICS/IXIT shall indicate how to present a correct value for accessCredentials ICS shall indicate how to request an incorrect value for an attributeId
7	Send GET.request	Receive GET.response containing (GET.response eid = n1 attributeList returnCode OPTIONAL)	 Same as in GET.request The ICS shall indicate whether attributes are present or not due to an incorrect attributeId in the request ICS/IXIT shall indicate whether returnCode is present and the value

Table C.15 (*continued*)

Step	Action	Expected response	Note
8	Configure a GET.request in confirmed mode (Mode = True eid = n2 accessCredentials = OCTET STRING attributeIdList)		 Mode is always true ICS shall indicate an incorrect value for eid ICS/IXITshall indicate how to present a correct value for accessCredentials ICS shall indicate correct values for attributeIds
9	Send GET.request	Receive GET.response containing (GET.response eid = n1 attributeList returnCode OPTIONAL)	 Same as in GET.request The ICS shall indicate whether an attributeList is present or not ICS/IXIT shall indicate whether returnCode is present and the value

C.1.3.1.3 Acceptance criteria

Table C.16 — TC03-A acceptance criteria

Item	Acceptance Criteria	Limit	Note
1	GET.response is sent from the OBU	with a complete attributeList in step 3 without any attribute values in step 5 without or with an incomplete attribute list in step 7 without any attribute values in step 9	The returnCode shall be as used according to ICS/IXIT

C.1.4 TC04 DSRC L7 SET

C.1.4.1 TC04-A basic DSRC L7 set

C.1.4.1.1 Equipment and set-up

The test set-up shall be according to what is described in Clause 6.

OBUs according to Clause 5 shall be used. The supplier's ICS and IXIT shall include information concerning:

— set of correct (n1) and incorrect (n2) value for eid;

— set of correct and incorrect values for eid, accessCredentials (if implemented) and attributeList in the SET request test frames;

— set of expected values for returnCode (if implemented) in the SET response test frames related to above possible correct/incorrect values for eid, accessCredentials and attributeList;

— algorithms and master keys values related to secure services inherent in the protocol (if implemented); and

— layer 2 services that are used for transmission of the Layer 7 APDUs.

The equipment shall be placed in such a way that optimal communication can be foreseen.

C.1.4.1.2 Main execution steps

Main execution steps include:

— step 1: Initialization;

— steps 2-3: Confirmed mode, Eid; correct value, AccessCredentials; correct value, AttributeList; correct values;

— steps 4-5: Confirmed mode, Eid; correct value, AccessCredentials; incorrect value, AttributeList, correct values;

— steps 6-7: Confirmed mode, Eid; correct value, AccessCredentials; correct value, AttributeList; non-existing attribute; and

— steps 8-9: Confirmed mode, Eid; non-existing element, AccessCredentials; correct value, AttributeList; correct value.

Table C.17 — Execution steps for TC04-A basic DSRC L7 SET

Step	Action	Expected response	Note
1	Perform Initialization according to TC-01 steps 1-2		
2	Configure a SET.request in confirmed mode (mode = True eid = n1 accessCredentials = OCTET STRING attributeList)		 ICS shall indicate a correct value for eid If present, ICS/IXIT shall indicate how to present a correct value for accessCredentials ICS shall indicate correct values for attributes
3	Send SET.request	Receive SET.response containing;(eid = n1 returnCode OPTIONAL)	 Same as in SET.request ICS shall indicate whether returnCode is present
4	Configure a SET.request in confirmed mode (Mode = True eid = n1 accessCredentials = OCTET STRING attributeList)		 ICS shall indicate a correct value for eid If present, ICS/IXIT shall indicate how to present an incorrect value for accessCredentials ICS shall indicate correct values for attributes
5	Send SET.request	Receive SET.response containing;(eid = n1 returnCode OPTIONAL)	 Same as in SET.request ICS/IXIT shall indicate how this situation is handled and how returnCode is used

Table C.17 (*continued*)

Step	Action	Expected response	Note
6	Configure a SET.request in confirmed mode (mode = True eid = n1 accessCredentials = OCTET STRING attributeList)		ICS shall indicate a correct value for eid If present, ICS/IXIT shall indicate how to present a correct value for accessCredentials ICS shall indicate how to present an incorrect value for an attribute
7	Send SET.request	Receive SET.response containing;(eid = n1 returnCode OPTIONAL)	Same as in SET.request ICS/IXIT shall indicate how this situation is handled and how returnCode is used
8	Configure a SET.request in confirmed mode (mode = True eid = n2 accessCredentials = OCTET STRING attributeList)		ICS shall indicate an incorrect value for eid If present, ICS/IXIT shall indicate how to present a correct value for accessCredentials ICS shall indicate correct values for attributes
9	Send SET.request	Receive SET.response containing;(eid = n2 returnCode OPTIONAL)	Same as in SET.request ICS/IXIT shall indicate how this situation is handled and how returnCode is used

C.1.4.1.3 Acceptance criteria

Table C.18 — TC04-A acceptance criteria

Item	Acceptance criteria	Limit	Note
1	SET.response is sent from the OBU	step 3 step 5 step 7 step 9	Check that the OBU handles the different cases and uses the returnCode according to ICS/IXIT

C.2 Basic EFC functions test suite

C.2.1 TC10 EFC GET STAMPED

C.2.1.1 TC10-A basic EFC GET STAMPED

C.2.1.1.1 Equipment and set-up

The test set-up shall be according to what is described in Clause 6.

OBUs according to Clause 5 shall be used. The supplier's ICS and IXIT shall include information concerning:

— set of correct and incorrect values for eid, accessCredentials (if implemented), attributeIdList and keyReference in the ACTION request test frames;

— set of expected values for responseParameter and returnCode (if implemented) in the ACTION response test frames related to above possible correct/incorrect values for eid, accessCredentials and attributeIdList;

— algorithms and master keys values related to secure services inherent in the protocol (if implemented);

— layer 2 services that are used for transmission of the layer 7 APDUs.

The equipment shall be placed in such a way that optimal communication can be foreseen.

C.2.1.1.2 Main execution steps

Main execution steps include:

— step 1: Initialization;

— steps 2-3: Eid; correct value, AccessCredentials; correct value, AttributeIdList; correct values, KeyRef; correct value;

— steps 4-5: Eid; correct value, AccessCredentials; incorrect value, AttributeIdList, correct values, KeyRef; correct value;

— steps 6-7: Eid; correct value, AccessCredentials; correct value, AttributeIdList; non-existing attribute, KeyRef; correct value;

— steps 8-9: Eid; non-existing element, AccessCredentials; correct value, AttributeIdList; correct value, KeyRef; correct value; and

— steps 10-11: Eid; correct value, AccessCredentials; correct value, AttributeIdList; correct values, KeyRef; incorrect value.

Table C.19 — Execution steps for TC10-A basic EFC GET STAMPED

Step	Action	Expected response	Note
1	Perform Initialization according to TC-01 steps 1-2		
2	Configure an ACTION.request in confirmed mode (mode = True eid = n1 actionType = 0 accessCredentials = OCTET STRING attributeIdList keyRef = K_1)		Mode is always true ICS shall indicate a correct value for eid GET_STAMPED If present, ICS/IXIT shall indicate how to present a correct value for accessCredentials ICS shall indicate correct values for attributeIds ICS/IXIT shall indicate correct values for keyRef
3	Send ACTION.request	Receive ACTION.response containing;(eid = n1 attributeList authenticator returnCode OPTIONAL)	Same as in ACTION.request requested attributes ICS/IXIT shall specify how the authenticator is calculated ICS shall indicate whether returnCode is present

Table C.19 (*continued*)

Step	Action	Expected response	Note
4	Configure an ACTION.request in confirmed mode (mode = True eid = n1 actionType = 0 accessCredentials = OCTET STRING attributeIdList keyRef = K_1)		 Mode is always true ICS shall indicate a correct value for eid GET_STAMPED If present, ICS/IXIT shall indicate how to present an incorrect value for accessCredentials ICS shall indicate correct values for attributeIds ICS shall indicate correct values for keyRef
5	Send ACTION.request	Receive ACTION.response containing;(eid = n1 attributeList authenticator returnCode OPTIONAL)	 Same as in ACTION.request requested attributes not present due to incorrect accessCredentials ICS/IXIT shall specify the content in the authenticator ICS/IXIT shall indicate whether returnCode is present and its value

Table C.19 (*continued*)

Step	Action	Expected response	Note
6	Configure an ACTION.request in confirmed mode (mode = True eid = n1 actionType = 0 accessCredentials = OCTET STRING attributeIdList keyRef = K_1)		 Mode is always true ICS shall indicate a correct value for eid GET_STAMPED If present, ICS shall indicate how to present a correct value for accessCredentials ICS shall indicate an incorrect value for attributeIds ICS shall indicate correct values for keyRef
7	Send ACTION.request	Receive ACTION.response containing;(eid = n1 attributeList authenticator returnCode OPTIONAL)	 Same as in ACTION.request The ICS shall indicate whether attributes are present or not due to an incorrect attributeId in the request ICS/IXIT shall specify how the authenticator is calculated ICS/IXIT shall indicate whether returnCode is present and its value

Table C.19 (*continued*)

Step	Action	Expected response	Note
8	Configure an ACTION.request in confirmed mode (mode = True eid = n2 actionType = 0 accessCredentials = OCTET STRING attributeIdList keyRef = K_1)		 Mode is always true ICS shall indicate an incorrect value for eid GET_STAMPED If present, ICS/IXIT shall indicate how to present a correct value for accessCredentials ICS shall indicate a correct value for attributeIds ICS shall indicate correct values for keyRef
9	Send ACTION.request	Receive ACTION.response containing;(eid = n2 attributeList authenticator returnCode OPTIONAL)	 Same as in ACTION.request The ICS shall indicate whether an attributeList is present or not ICS shall specify whether the authenticator is present or not ICS/IXIT shall indicate whether returnCode is present and its value

Table C.19 (*continued*)

Step	Action	Expected response	Note
10	Configure an ACTION.request in confirmed mode (mode = True eid = n1 actionType = 0 accessCredentials = OCTET STRING attributeIdList keyRef = K_2)		 Mode is always true ICS shall indicate a correct value for eid GET_STAMPED If present, ICS/IXIT shall indicate how to present a correct value for accessCredentials ICS shall indicate a correct value for attributeIds ICS shall indicate an incorrect value for keyRef
11	Send ACTION.request	Receive ACTION.response containing;(eid = n1 attributeList authenticator returnCode OPTIONAL)	 Same as in ACTION.request The ICS shall indicate whether an attributeList is present or not ICS shall specify whether the authenticator is present or not ICS/IXIT shall indicate whether returnCode is present and its value

C.2.1.1.3 Acceptance criteria

Table C.20 — TC10-A acceptance criteria

Item	Acceptance criteria	Limit	Note
1.	ACTION.response is sent from the OBU	with a complete attributeList and a correct authenticator in step 3 without any attribute values in step 5 without or with an incomplete attribute list in step 7 without any attribute values in step 9 without any attribute values in step 11	The Return Code shall be is used according to ICS/IXIT

C.2.2 TC12 EFC GET SECURE

C.2.2.1 TC12-A Basic EFC GET SECURE

C.2.2.1.1 Equipment and set-up

The test set-up shall be according to what is described in Clause 6.

OBUs according to Clause 5 shall be used. The supplier's ICS and IXIT shall include information concerning:

— set of correct and incorrect values for eid, accessCredentials (if implemented) and actionParameter in the ACTION request test frames;

— set of expected values for responseParameter and returnCode (if implemented) in the ACTION response test frames related to above possible correct/incorrect values for eid, accessCredentials and actionParameter;

— algorithms and master keys values related to secure services inherent in the protocol (if implemented); and

— layer 2 services that are used for transmission of the layer 7 APDUs.

The equipment shall be placed in such a way that optimal communication can be foreseen.

C.2.2.1.2 Main execution steps

Main execution steps comprise tests concerning handling of:

— step 1: Initialization of OBU;

— steps 2-3: Eid; incorrect value, AccessCredentials; correct value, ActionParameter; correct value;

— steps 4-5: Eid; correct value, AccessCredentials; incorrect value, ActionParameter; correct value;

— steps 6-7: Eid; correct value, AccessCredentials; correct value, ActionParameter; incorrect value; and

— steps 8-9: Eid; correct value, AccessCredentials; correct value, ActionParameter; correct value.

Table C.21 — Execution steps for TC12-A basic EFC GET SECURE

Step	Action	Expected response	Note
1	Perform Initialization according to TC-01 steps 1-2		
2	Configure an ACTION.request in confirmed mode (mode = True eid = n2 actionType = 2 accessCredentials = OCTET STRING actionParameter = Container)		Mode is always true ICS shall indicate an incorrect value for eid GET_SECURE If present, ICS/IXIT shall indicate how to present a correct value for accessCredentials ICS shall indicate a correct value for actionParameter
3	Send ACTION.request	Receive ACTION.response containing;(eid = n2 responseParameter PRESENT returnCode OPTIONAL)	Same as in ACTION.request ICS/IXIT shall specify the content of the responseParameter ICS/IXIT shall indicate whether returnCode is present and the value

Table C.21 (*continued*)

Step	Action	Expected response	Note
4	Configure an ACTION.request in confirmed mode (mode = True eid = n1 actionType = 2 accessCredentials = OCTET STRING actionParameter = Container)		Mode is always true ICS shall indicate a correct value for eid GET_SECURE If present, ICS/IXIT shall indicate how to present an incorrect value for accessCredentials ICS shall indicate a correct value for actionParameter
5	Send ACTION.request	Receive ACTION.response containing;(eid = n1 responseParameter PRESENT returnCode OPTIONAL)	Same as in ACTION.request ICS/IXIT shall specify the content of the responseParameter ICS/IXIT shall indicate whether returnCode is present and its value
6	Configure an ACTION.request in confirmed mode (mode = True eid = n1 actionType = 2 accessCredentials = OCTET STRING actionParameter = Container)		Mode is always true ICS shall indicate a correct value for eid GET_SECURE If present, ICS/IXIT shall indicate how to present a correct value for accessCredentials ICS shall indicate an incorrect value for actionParameter

Table C.21 (*continued*)

Step	Action	Expected response	Note
7	Send ACTION.request	Receive ACTION.response containing;(eid = n1 responseParameter PRESENT returnCode OPTIONAL)	 Same as in ACTION.request ICS/IXIT shall specify the content of the responseParameter ICS/IXIT shall indicate whether returnCode is present and the value
8	Configure an ACTION.request in confirmed mode (mode = True eid = n1 actionType = 2 accessCredentials = OCTET STRING actionParameter = Container)		 Mode is always true ICS shall indicate a correct value for eid GET_SECURE If present, ICS/IXIT shall indicate how to present a correct value for accessCredentials ICS shall indicate a correct value for actionParameter
9	Send ACTION.request	Receive ACTION.response containing;(eid = n1 responseParameter PRESENT returnCode OPTIONAL)	 Same as in ACTION.request ICS/IXIT shall specify the content of the responseParameter ICS/IXIT shall indicate whether returnCode is present

C.2.2.1.3 Acceptance criteria

Table C.22 — TC12-A acceptance criteria

Item	Acceptance criteria	Limit	Note
1	ACTION.response	'0001'B	
2	Eid	According to C.2.2.1.2 and ISO14906, 5.3.2	
3	responseParameter	According to C.2.2.1.2	ICS/IXIT shall specify valid responseParameter value limit
4	ReturnCode	According to EN 12834	ICS/IXIT shall specify how returnCode is implemented

C.2.3 TC13 EFC SET SECURE

C.2.3.1 TC13-A basic EFC GET SECURE

C.2.3.1.1 Equipment and set-up

The test set-up shall be according to what is described in Clause 6.

OBUs according to Clause 5 shall be used. The supplier's ICS and IXIT shall include information concerning:

— set of correct and incorrect values for eid, accessCredentials (if implemented) and actionParameter in the ACTION request test frames;

— set of expected values for responseParameter and returnCode (if implemented) in the ACTION response test frames related to above possible correct/incorrect values for eid, accessCredentials and actionParameter;

— algorithms and master keys values related to secure services inherent in the protocol (if implemented); and

— layer 2 services that are used for transmission of the layer 7 APDUs.

The equipment shall be placed in such a way that optimal communication can be foreseen.

C.2.3.1.2 Main execution steps

Main execution steps comprise tests concerning handling of:

— step 1: Initialization of OBU;

— steps 2-3: Confirmed mode: Eid; incorrect value, AccessCredentials; correct value, ActionParameter; correct value;

— steps 4-5: Confirmed mode: Eid; correct value, AccessCredentials; incorrect value, ActionParameter; correct value;

— steps 6-7: Confirmed mode: Eid; correct value, AccessCredentials; correct value, ActionParameter; incorrect value; and

— steps 8-9: Confirmed mode: Eid; correct value, AccessCredentials; correct value, ActionParameter; correct value.

Table C.23 — Execution steps for TC13-A basic EFC GET SECURE

Step	Action	Expected response	Note
1	Perform Initialization according to TC-01 steps 1-2		
2	Configure an ACTION.request in confirmed mode (mode = True eid = n2 actionType = 3 accessCredentials = OCTET STRING actionParameter = Container)		confirmed mode ICS shall indicate an incorrect value for eid SET_SECURE If present, ICS/IXIT shall indicate how to present a correct value for accessCredentials ICS shall indicate a correct value for actionParameter
3	Send ACTION.request	Receive ACTION.response containing;(eid = n2 responseParameter OPTIONAL returnCode OPTIONAL)	 Same as in ACTION.request The ICS/IXIT shall specify the content of the responseParameter ICS/IXIT shall indicate whether returnCode is present and the value
4	Configure an ACTION.request in confirmed mode (mode = True eid = n1 actionType = 3 accessCredentials = OCTET STRING actionParameter = Container)		 confirmed mode ICS shall indicate a correct value for eid SET_SECURE If present, ICS/IXIT shall indicate how to present an incorrect value for accessCredentials ICS shall indicate a correct value for actionParameter

Table C.23 (*continued*)

Step	Action	Expected response	Note
5	Send ACTION.request	Receive ACTION.response containing;(eid = n1 responseParameter OPTIONAL returnCode OPTIONAL)	 Same as in ACTION.request ICS/IXIT shall specify the content of the responseParameter ICS/IXIT shall indicate whether returnCode is present and the value
6	Configure an ACTION.request in confirmed mode (mode = True eid = n1 actionType = # accessCredentials = OCTET STRING actionParameter = Container)		 confirmed mode ICS shall indicate a correct value for eid SET_SECURE If present, ICS/IXIT shall indicate how to present a correct value for accessCredentials ICS shall indicate an incorrect value for actionParameter
7	Send ACTION.request	Receive ACTION.response containing;(eid = n1 responseParameter OPTIONAL returnCode OPTIONAL)	 Same as in ACTION.request ICS/IXIT shall specify the content of the responseParameter ICS/IXIT shall indicate whether returnCode is present and the value

Table C.23 (*continued*)

Step	Action	Expected response	Note
8	Configure an ACTION.request in confirmed mode (mode = True eid = n1 actionType = 3 accessCredentials = OCTET STRING actionParameter = Container)		confirmed mode ICS shall indicate a correct value for eid SET_SECURE If present, ICS/IXIT shall indicate how to present a correct value for accessCredentials ICS shall indicate a correct value for actionParameter
9	Send ACTION.request	Receive ACTION.response containing;(eid = n1 responseParameter OPTIONAL returnCode OPTIONAL)	Same as in ACTION.request ICS/IXIT shall specify the content of the responseParameter ICS/IXIT shall indicate whether returnCode is present

C.2.3.1.3 Acceptance criteria

Table C.24 — TC14-A acceptance criteria

Item	Acceptance criteria	Limit	Note
1	ACTION.response	'0001'B	
2	Eid	According to C.2.3.1.2 and ISO 14906, 5.3.2	
3	ResponseParameter	According to C.2.3.1.2	ICS/IXIT shall specify content of responseParameter

C.2.4 TC20 EFC SET MMI

C.2.4.1 TC20-A basic EFC SET MMI

C.2.4.1.1 Equipment and set-up

The test set-up shall be according to what is described in Clause 6.

OBUs according to Clause 5 shall be used. The supplier's ICS and IXIT shall include information concerning:

— set of correct and incorrect values for actionParameter and in the ACTION request test frames;

— set of expected values for returnCode (if implemented) in the ACTION response test frames related to above possible correct/incorrect value for the actionParameter; and

— layer 2 services that are used for transmission of the layer 7 APDUs.

The equipment shall be placed in such a way that optimal communication can be foreseen.

C.2.4.1.2 Main execution steps

The main execution steps include:

— step 1: Initialization;

— steps 2-3: Non-confirmed mode; ActionParameter "correct";

— steps 4-5: Confirmed mode; ActionParameter "correct";

— steps 6-7: Confirmed Mode; ActionParameter "not correct".

Table C.25 — Execution steps for TC20-A basic EFC SET MMI

Step	Action	Expected response	Note
1	Perform Initialization according to TC-01 steps 1-2		
2	Configure an ACTION.req in non-confirmed mode (action-Request mode = False eid = 0 actionType = 10 actionParameter)		 non-confirmed mode SET_MMI declaration shall indicate a correct value for actionParameter
3	Send frame as specified in step 2	None	Check that the OBU responds correctly to the EFC function

Table C.25 (*continued*)

Step	Action	Expected response	Note
4	Configure an ACTION.req in confirmed mode (action-Request mode = True eid = 0 actionType = 10 actionParameter)		confirmed mode declaration shall indicate a correct value for actionParameter
5	Send frame as specified in step 4	ACTION.rs, eid = 0, returnCode)	ICS/IXIT shall indicate what returnCode is present Check that the OBU responds correctly to the EFC function
6	Configure an ACTION.req in confirmed mode (action-Request mode = True eid = 0 actionType = 10 actionParameter)		confirmed mode declaration shall indicate an incorrect value for actionParameter
7	Send frame as specified in step 6	ACTION.rs, eid = 0, returnCode)	ICS/IXIT shall indicate what returnCode is present

C.2.4.1.3 Acceptance criteria

Table C.26 — TC20-A acceptance criteria

Item	Acceptance Criteria	Limit	Note
1	ACTION.response is not sent from the OBU	in step 3	The returnCode shall be according to ICS/IXIT The OBU shall react correctly on the EFC function
2	ACTION.response is sent from the OBU	with a returnCode indication success in step 5 with a returnCode indicating error in step 7	ICS/IXIT shall specify how returnCode is implemented

C.2.5 TC25 EFC ECHO

C.2.5.1 TC25-A Basic EFC ECHO

C.2.5.1.1 Equipment and set-up

The test set-up shall be according to what is described in Clause 6.

OBUs according to Clause 5 shall be used. The supplier's ICS and IXIT shall include information concerning:

— set of correct and incorrect values for actionParameter and in the ACTION request test frames;

— set of expected values for responseParameter and returnCode (if implemented) in the ACTION response test frames related to above possible correct/incorrect value for the actionParameter; and

— layer 2 services that are used for transmission of the Layer 7 APDUs.

The equipment shall be placed in such a way that optimal communication can be foreseen.

C.2.5.1.2 Main execution steps

Main execution steps include:

— step 1: Initialization;

— steps 2-3: Non-confirmed mode; ActionParameter “correct”;

— steps 4-5: Confirmed mode; ActionParameter “correct”;

— steps 6-7: Confirmed Mode; ActionParameter “not correct”.

Table C.27 — Execution steps for TC25-A basic EFC ECHO

Step	Action	Expected response	Note
1	Perform Initialization according to TC-01 steps 1-2		
2	Configure an ACTION.req in non-confirmed mode (action-Request mode = False eid = 0 actionType = 15 actionParameter)		non-confirmed mode ECHO declaration shall indicate a correct value for actionParameter
3	Send frame as specified in step 5	None	
4	Configure an ACTION.req in confirmed mode (action-Request mode = True eid = 0 actionType = 15 actionParameter)		confirmed mode declaration shall indicate a correct value for actionParameter
5	Send frame as specified in step 4	ACTION.rs, eid = 0, responseParameter returnCode)	same as actionParameter ICS/IXIT shall indicate what returnCode is present

Table C.27 (*continued*)

Step	Action	Expected response	Note
6	Configure an ACTION.req in confirmed mode (action-Request mode = True eid = 0 actionType = 15 actionParameter)		 confirmed mode declaration shall indicate an incorrect value for actionParameter
7	Send frame as specified in step 4	ACTION.rs, eid = 0, responseParameter returnCode)	 ICS/IXIT shall indicate what returnCode is present

C.2.5.1.3 Acceptance criteria

Table C.28 — TC25-A acceptance criteria

Item	Acceptance criteria	Limit	Note
1	ACTION.response is not sent from the OBU	in step 3	The returnCode shall be as used according to ICS/IXIT
2	ACTION.response is sent from the OBU	with a correct responseParameter in step 3 without a correct responseParameter in step 7	The returnCode shall be as used according to ICS/IXIT

C.3 EFC transaction test suite

C.3.1 TC30 EFC TRANSACTION FUNCTIONALITY

C.3.1.1 TC30-A CARDME transaction

The purpose of the test is to verify the OBU conformance to the CARDME transaction as defined in Annex B of ISO 14906.

C.3.1.1.1 Equipment and set-up

The test set-up shall be according to what is described in Clause 6.

OBUs according to Clause 5 shall be used. The supplier's ICS and IXIT shall include information concerning:

— layer 2 services that are used for transmission of the layer 7 APDUs;

— what profile(s) that are supported; and

— the implementation of the CARDME transaction.

In this test case, it is assumed that:

— layer 2 services according to CEN EN 12795 are implemented in the OBU; and

— the OBU supports profiles P_a and P_b.

The equipment shall be placed in such a way that optimal communication can be foreseen.

C.3.1.1.2 Main execution steps

Main execution steps comprise tests concerning:

— steps 1-2: Initialization phase;

— step 3: Presentation phase;

— step 4: Presentation phase for foreign OBEs;

— step 5: Receipt phase;

— step 6: Echo phase; and

— step 7: Closing phase.

Table C.29 — Execution steps for TC30-A CARDME transaction

Step	Action	Expected response	Note
1	Configure the BST according to Annex B.3.1.1 in prEN ISO 14906 with: profile = P_a mandatory applications = EFC beaconId manufacturerid = M_a IndividualId = 1		It is assumed that the OBU has not received any BST with this beaconId during the last 255 s
2	Send BST every 5-10 ms (until initialization is performed and a VST is received)	VST according to B.3.1.4 in ISO 14906	t-ref is the time in the BST that was answered by the VST
3	Perform the presentation phase by invocation GET_STAMPED.request concatenated with GET.request in confirmed mode (mode = TRUE) according to B.3.2.1 of ISO 14906	Response to the request according to B.3.2.2 of ISO 14906	
4	Perform the presentation phase for foreign OBE by invocation of GET_STAMPED.request in confirmed mode (mode = TRUE) according to B.3.3.1 of ISO 14906	Response to the request according to B.3.3.2 of ISO 14906	
5	Perform the receipt phase by invocation of SET.request concatenated with and SET_MMI in confirmed mode (mode = TRUE) according to B.3.4.1 of ISO 14906	Response to the request according to B.3.4.2 of ISO 14906	
6	Perform the echo phase by invocation of ECHO.request in confirmed mode (mode = TRUE) according to B.3.4.1 of ISO 14906	Response to the request according to B.3.5.2 of ISO 14906	
7	Perform the closing phase by invocation of EVENT_REPORT (Release) in non-confirmed mode (mode = FALSE) according to B.3.5.3 of ISO 14906	No response	