PD CEN ISO/TR 16401-1:2018

BSI Standards Publication

Electronic fee collection - Evaluation of equipment for conformity to ISO/TS 17575-2

Part 1: Test suite structure and test purposes (ISO/TR 16401-1:2018)

Documents
Document Sharing Hub

Full document details: PD CEN ISO/TR 16401-1:2018

National foreword

This Published Document is the UK implementation of CEN ISO/TR 16401-1:2018. It is identical to ISO/TR 16401-1:2018. It supersedes PD CEN ISO/TS 16401-1:2012, which is withdrawn.

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

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

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

© The British Standards Institution 2018
Published by BSI Standards Limited 2018

ISBN 978 0 580 99006 9

ICS 03.220.20; 35.240.60

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

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 31 March 2018.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: PD CEN ISO/TR 16401-1:2018

TECHNICAL REPORT

RAPPORT TECHNIQUE

TECHNISCHER BERICHT

CEN ISO/TR 16401-1

February 2018

ICS 03.220.20; 35.240.60

Supersedes CEN ISO/TS 16401-1:2012

English Version

Electronic fee collection - Evaluation of equipment for conformity to ISO/TS 17575-2 - Part 1: Test suite structure and test purposes (ISO/TR 16401-1:2018)

Perception du télépéage - Évaluation de conformité de l'équipement à l'ISO/TS 17575-2 - Partie 1: Structure de la suite d'essais et objectifs d'essai (ISO/TR 16401-1:2018)

Elektronische Gebührenerhebung - Konformitätsevaluierung von Einrichtungen nach ISO/TS 17575-2 - Teil 1: Struktur und Zweck des Prüfprogrammes (ISO/TR 16401-1:2018)

This Technical Report was approved by CEN on 14 January 2018. It has been drawn up by the Technical Committee CEN/TC 278.

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

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

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

© 2018 CEN All rights of exploitation in any form and by any means reserved worldwide for CEN national Members.

Ref. No. CEN ISO/TR 16401-1:2018 E

Documents
Document Sharing Hub

Full document details: PD CEN ISO/TR 16401-1:2018

European Foreword

This document (CEN ISO/TR 16401-1:2018) has been prepared by Technical Committee ISO/TC 204 "Intelligent transport systems" in collaboration with Technical Committee CEN/TC 278 “Intelligent transport systems” the secretariat of which is held by NEN.

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

This document supersedes CEN ISO/TS 16401-1:2012.

This document has been prepared under a mandate given to CEN by the European Commission and the European Free Trade Association.

Endorsement notice

The text of ISO/TR 16401-1:2018 has been approved by CEN as CEN ISO/TR 16401-1:2018 without any modification.

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

Contents

Page

Foreword ..... **iv**
Introduction ..... **v**
1 Scope **1**
2 Normative references **1**
3 Terms and definitions **1**
4 Abbreviated terms **4**
5 Test Suite Structure **5**
5.1 Structure 5
5.2 Reference to conformance test specifications 5
5.3 Test purposes (TP) 5
5.3.1 TP definition conventions 5
5.3.2 TP naming conventions 6
5.4 Protocol Conformance Test Report (PCTR) 7
Annex A (informative) **Test purposes (TP) for Front End Communications API** **8**
Annex B (informative) **Test purposes (TP) for Front End Application** **137**
Annex C (informative) **PCTR proforma for Front End Communications API** **141**
Annex D (informative) **PCTR proforma for Front End Application** **148**
Bibliography **152**

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

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 procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

This edition of ISO/TR 16401-1 cancels and replaces ISO/TS 16401-1:2012, which has been technically revised.

The main changes compared to the previous edition are as follows:

— the document has been converted from a Technical Specification to a Technical Report;

— the terms and definitions have been revised;

— the test purpose naming convention has been changed, i.e. “/” has been replaced by “_”;

— editorial corrections, as well as changes to improve readability have been made.

A list of all parts in the ISO/TR 16401 series can be found on the ISO website.

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

Introduction

This document is part of a set of standards that supports interoperability of autonomous electronic fee collection (EFC) systems. Autonomous systems use satellite positioning, often combined with additional sensor technologies such as gyroscopes, odometers and accelerometers, to localize the vehicle and to find its position on a map containing the charged geographic objects, such as charged roads or charged areas. From the charged objects, the vehicle characteristics, the time of day and other data that are relevant for describing road use, the tariff and ultimately, the road usage fee is determined.

The ISO/TR 16401 series provides tests to assess the Front End Communications API and Front End Application behaviours compliancy towards the requirements listed in ISO 17575-2. This document contains the definition of such tests in the form of test purposes, listing the initial conditions, references and individual steps in a structured textual manner. ISO/TR 16401-2 contains the identical tests written in Testing and Test Control Notation version 3 (TTCN v3).

Autonomous on-board equipment (OBE) operates without relying on dedicated roadside infrastructure by employing wide-area technologies such as Global Navigation Satellite Systems (GNSS) and Cellular Communications Networks (CN). Therefore, autonomous systems can also be referred to as GNSS/CN systems.

ISO/TR 16401-1 is based on

— ISO 17575-2, and

— the ISO 9646 family of standards on conformance test methodology.

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

Electronic fee collection — Evaluation of equipment for conformity to ISO/TS 17575-2 —

Part 1:
Test suite structure and test purposes

1 Scope

This document covers the test purposes for Front End Communications API covering functionalities related to instance handling, session handling, communication service primitives (i.e. sending/receiving of ADUs) and visible state transitions. It covers EFC communication services described in ISO 17575-2:2016, Clause 5 and PICS proforma in ISO 17575-2:2016, B.2. Claims related to Front End storage capacity are out of scope of this document.

This document covers the test purposes for Front End Application related to session establishment on Back End request and related to session re-establishment when session requested by Back End failed. There are no other claims with respect to Front End Application described in ISO 17575-2.

The underlying communication technology requirements for layer 1 to 4 specified in ISO 17575-2:2016, Clause 6 are out of scope of this document.

Similarly, Back End Communications API is out of scope of this document. According to ISO 17575-2 it is expected that these Front End Communications API will be “reflected” in the BE; however, BE Communications API is out of scope of ISO 17575-2.

Test purposes have been organized into the test suite groups, designated for the Front End Communications API and Front End Application, respectively.

Aside from the test purposes, this document also provides proforma conformance test reports templates for both the Front End and Back End test purposes.

ISO 17575-2 contains more information regarding the requirements against which the conformance is evaluated in this document.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

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

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at https://www.electropedia.org/

— ISO Online browsing platform: available at https://www.iso.org/obp

3.1
area charging
charging based on road usage within a given area

[SOURCE: ISO 17575-1:2016, 3.1]

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

3.2
attribute
addressable package of data consisting of a single *data element* (3.9)or structured sequences of data elements

[SOURCE: ISO 17575-1:2016, 3.2]

3.3
authenticator
data, possibly encrypted, that is used for authentication

[SOURCE: EN 15509:2014, 3.3]

3.4
Back End
part of a back office system interfacing to one or more *Front Ends* (3.11)

[SOURCE: ISO 17575-1:2016, 3.4]

3.5
charge object
geographic or road related object for the use of which a charge is applied

[SOURCE: ISO 17575-1:2016, 3.5]

3.6
charge report
information containing road usage and related information originated at the *Front End* (3.11)

[SOURCE: ISO 17575-1:2016, 3.6]

3.7
cordon
border line of an area

[SOURCE: ISO 17575-1:2016, 3.7]

3.8
cordon charging
charging for the crossing of a *cordon* (3.7)

[SOURCE: ISO 17575-1:2016, 3.8]

3.9
data element
coded information, which might itself consist of lower level information structures

[SOURCE: ISO 17575-1:2016, 3.9]

3.10
data set
logical set of *data elements* (3.9) with a semantic relation

[SOURCE: ISO 17575-3:2016, 3.10]

3.11
Front End
part of a tolling system consisting of an *OBE* (3.14) and possibly a *proxy* (3.15) where road tolling information and usage data are collected and processed for delivery to the *Back End* (3.4)

[SOURCE: ISO/TS 19299:2015, 3.17]

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

3.12
Front End Application
part of the *Front End* (3.11) above the API

[SOURCE: ISO 17575-2:2016, 3.12]

3.13
layout
technical description of the location of tolled objects including their borders

[SOURCE: ISO 17575-3:2016, 3.12]

3.14
on-board equipment
OBE
all required equipment on-board a vehicle for performing required EFC functions and communication services

3.15
proxy
optional part of a *Front End* (3.11) that communicates with external equipment and processes the data received into an agreed format to be delivered to the *Back End* (3.4)

[SOURCE: ISO 17575-1:2016, 3.13]

3.16
road section charging
tolling principle where the fee is due if predefined sections of roads are used

[SOURCE: ISO 17575-1:2016, 3.14]

3.17
toll
charge, tax or duty levied in connection to using a vehicle in a *toll domain* (3.21)

[SOURCE: ISO/TS 19299:2015, 3.42 modified]

3.18
tolled area
geographic area where a *toll* (3.17) is charged for road usage

[SOURCE: ISO 17575-3:2016, 3.17]

3.19
toll context
logical view as defined by *attributes* (3.2) and functions of the basic elements of a toll scheme consisting of a single basic tolling principle, a spatial distribution of the *charge objects* (3.5) and a single behaviour of the related *Front End* (3.11)

[SOURCE: ISO 17575-1:2016, 3.17]

3.20
toll context data
information defined by the responsible Toll Charger necessary to establish the *toll* (3.17) due for using a vehicle on a particular *toll context* (3.19) and to conclude the toll transaction

[SOURCE: ISO 12855:2015, 3.15]

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

3.21
toll domain
area or part of a road network where a certain *toll regime* (3.22) is applied

[SOURCE: ISO 17573:2010, 3.18]

3.22
toll regime
set of rules, including enforcement rules, governing the collection of *toll* (3.17) in a *toll domain* (3.21)

[SOURCE: ISO 17573:2010, 3.20]

3.23
toll scheme
organizational view of a *toll regime* (3.22), including the actors and their relationships

[SOURCE: ISO 17575-3:2016, 3.22]

3.24
transaction
whole of the exchange of information between two physically separated communication facilities

[SOURCE: ISO 17575-1:2016, 3.21]

3.25
transaction model
functional model describing the structure of electronic payment transactions

[SOURCE: ISO 14906:2011, 3.25 modified]

4 Abbreviated terms

ADU Application data unit

API Application Programming Interface

ASN.1 Abstract Syntax Notation One

ATS Abstract Test Suite

BE Back End

BI Behaviour invalid

BV Behaviour valid

CN Cellular network

IUT Implementation under test

EFC Electronic fee collection

FE Front End

GNSS Global Navigation Satellite Systems

ID Identifier

OBE On-board equipment

PCTR Protocol Conformance Test Report

PICS Protocol Implementation Conformance Statements

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP Test purposes

TSS Test Suite Structure

TTCN Testing and Test Control Notation

5 Test Suite Structure

5.1 Structure

Table 1 shows the Test Suite Structure (TSS).

Table 1 — Test Suite Structures

Group	Type of IUT	Behaviour
Instance handling	Front End Communications API	Behaviour valid
		Behaviour invalid
Session handling	Front End Communications API	Behaviour valid
		Behaviour invalid
	Front End Application	Behaviour valid
Communication service primitives	Front End Communications API	Behaviour valid
		Behaviour invalid
State transitions	Front End Communications API	Behaviour valid

5.2 Reference to conformance test specifications

This document takes into account already defined test purposes for conformance to the base standards by referencing them, so that

a) for test purposes that are **identical** to those defined in this document or the base standards conformance test cases, direct reference is reported; for reader's convenience, the title or a verbal description of the referenced test purpose is given, together with the reference,

b) for test purposes that are **derived** from those defined in the base standards conformance test cases, a direct reference is reported, plus an indication on how the referred test purpose has to be modified for the profile conformance testing,

c) for test purposes that are **specific** to ISO 17575-2, complete description is given, and

d) an indication on whether a test purpose is **identical**, **derived** or **specific** is given in each test purpose.

5.3 Test purposes (TP)

5.3.1 TP definition conventions

The TPs are defined following the rules shown in Table 2. Test purposes are defined in Annex A and Annex B, including the following special notation and symbol conventions.

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

Table 2 — TP definition rules

TP ID according to the TP naming conventions	**Title:**
	Reference:
	TP origin:
	Initial condition:
	Stimulus and expected behaviour:

TP ID	The TP ID is a unique identifier. It is specified according to the TP naming conventions defined in the subclause below.
Title	Short description of TP objective.
Reference	The reference should contain the references of the subject to be validated by the actual TP (specification reference, clause, paragraph), or the reference to the standard document defining the TP.
TP origin	Indicates if the TP is **identical** to a TP defined in another test standard, **derived** from a TP defined in another test standard, or **specific** for this standard profile.
Initial condition	The condition defines in which initial state the IUT has to be to apply the actual TP.
Stimulus and expected behaviour	Definition of the events the tester performs and the events that are expected from the IUT to conform to the base specification.

5.3.2 TP naming conventions

Each TP is given a unique identification. This unique identification is built up to contain the following string of information:

TP_<group>_<iut>_<x>_<nn>

TP : indicates that it is a test purpose;

<group> : indicates to which group the TP belongs;

<iut> : indicates the type of IUT, i.e. API or APPL;

<x> : indicates the type of testing, i.e. behaviour valid tests (BV) or behaviour invalid tests (BI);

<nn> : indicates the sequential TP number (01 to 99).

The naming conventions are as described in Table 3.

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

Table 3 — TP naming convention

Identifier: TP_<group>_<iut>_<x>-<nn>		
<group>		
applicable for FE Communications API	IH	Instance Handling
applicable for FE Communications API	SH	Session Handling
applicable for FE Application	SH	Session Handling
applicable for FE Communications API	CSP	Communications Service Primitives
applicable for FE Communications API	ST	State Transitions
<iut> = type of IUT	API	Front End Communications API
	APPL	Front End Application
<x> = type of testing	BV	Behaviour Valid Tests
	BI	Behaviour Invalid Tests
<nn> = sequential number	(01 to 99)	Test Purpose Number

5.4 Protocol Conformance Test Report (PCTR)

The supplier of the Front End is responsible for providing a conformance test report.

The supplier of the Front End Communications API should complete the Protocol Conformance Test Report (PCTR) proforma as defined in Annex C.

The supplier of the Front End Application should complete the Protocol Conformance Test Report (PCTR) proforma as defined in Annex D.

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

Annex A
(informative)

Test purposes (TP) for Front End Communications API

A.1 Overview

This annex contains the test purposes (TP) for the conformity evaluation of Front End Communications to ISO 17575-2.

A.2 TP symbols conventions

A special notation and symbol convention is used, as defined in this subclause.

Definitions of the symbols used in the description of the TPs are provided in Table A.1.

Table A.1 — Description of TP Symbols

Symbol	Description
XXX(Type1 = value1) ⇒	The tester executes an XXX method of FE Communications API with argument Type1 having a value of value1. Value1 is stored in the tester's memory for further TP processing.
⇐ R:ReturnedType	The IUT returns a value of type ReturnedType.
⇐ C:CallbackName (Type1)	The IUT provides a callback CallbackName receiving variable of type Type1.
Type ISO 17575-2	Anytime Type defined in ISO 17575-2 is used. It means a variable of Type.
A → B	A “is transformed” into B.
Ø	Means “empty” or “not set”.
A | B	A or B
A ! = B	A is not equal to B.
i = i + 1	Increment variable i by 1.

In testing the Front End Communications API, it is needed to trigger operations and observe the IUT feedback both from the Front End Application and remote end (i.e. Back End) perspective. Thus, there are two test points located as shown in Figure A.1.

Application emulation test point is used directly with the IUT and emulates the Front End Application layer. It is identified in the following test purposes by the AppEm discriminator.

Remote End emulation test point is linked with the IUT over communications channel. Depending on the test purposes, it emulates application, presentation and session layer. It is identified in the following test purposes by the RemEnd discriminator.

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

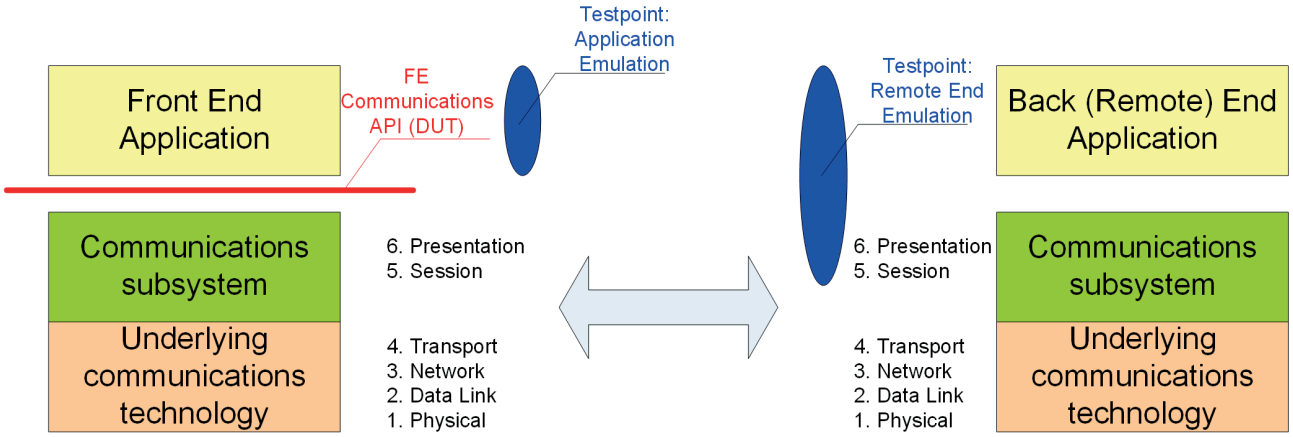

Figure A.1 — Handling of ADUs applicable for a particular TP

A.3 Instance handling

These test purposes apply to instance handling as described in ISO 17575-2:2016, B.2, with respect to following PICS proforma entries:

— API supports InitialiseInstance;

— API supports SetParameter;

— API supports GetParameter;

— API supports DeleteParameter;

— API supports DropInstance;

— API supports StackAvail.

A.3.1 BV test purposes (Behaviour Valid tests)

Test subgroup objective:

— to test IUT behaviour with respect to instance initialization, including multiple instance handling in parallel;

— to test IUT behaviour with respect to parameter setting and updating;

— to test IUT behaviour with respect to parameter getting;

— to test IUT behaviour with respect to parameter deleting;

— to test IUT behaviour with respect to availability of communications stack;

— to test IUT behaviour with respect to dropping the session with the following severities:

  — SENormal;

  — SEUrgent;

  — SEUnconditional.

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_01	**Verify the communications interface initialization.**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2.1			
Initial condition	FE Communications API shall handle at least one underlying communication stack: which StackID equals to stack1. Set of Callback instances is instantiated.			
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**InitialiseInstance** (StackID = stack1, Callbacks = cb1)	AppEm	⇒	
2		AppEm	⇐	**R: Instance**
3	Verify whether **Instance** is valid.			
4	IF verify OK THEN TP passed ELSE TP failed ENDIF			

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_02	**Verify the multiple instance communications interface initialization based on the same communications stack.**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.2.1
Initial condition	FE Communications API shall handle at least one underlying communication stack: which StackID equals to stack1. Sets of Callback instances are instantiated.

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**InitialiseInstance** (StackID = stack1, Callbacks = cb1)	AppEm	⇒	
2		AppEm	⇐	**R: Instance**
3	Verify whether **Instance** is valid.			
4	IF verify OK THEN GOTO step 5 ELSE TP failed ENDIF			
5	**InitialiseInstance** (StackID = stack1, Callbacks = cb2)	AppEm	⇒	
6		AppEm	⇐	**R: Instance**
7	Verify whether **Instance** is valid.			
8	IF verify OK THEN GOTO step 9 ELSE TP failed ENDIF			
9	**InitialiseInstance** (StackID = stack1, Callbacks = cb3)	AppEm	⇒	
10		AppEm	⇐	**R: Instance**
11	Verify whether **Instance** is valid.			
12	IF verify OK THEN TP passed ELSE TP failed ENDIF			

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_03	**Verify the multiple instance communications interface initialization based on different communications stack.**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.2.1
Initial condition	FE Communications API shall handle at least two underlying communication stacks: which StackID equals to stack1 and stack2. Sets of Callback instances are instantiated.

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**InitialiseInstance** (StackID = stack1, Callbacks = cb1)	AppEm	⇒	
2		AppEm	⇐	**R: Instance**
3	Verify whether **Instance** is valid.			
4	IF verify OK THEN GOTO step 5 ELSE TP failed ENDIF			
5	**InitialiseInstance** (StackID = stack2, Callbacks = cb2)	AppEm	⇒	
6		AppEm	⇐	**R: Instance**
7	Verify whether **Instance** is valid.			
8	IF verify OK THEN TP passed ELSE TP failed ENDIF			

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_04	**Verify that parameter is set by the Front End Application (single parameter).**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2.1			
Initial condition	A valid Instance, instance1, is created.			
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**SetParameter** (Instance = instance1, Parameter = "Parameter1", Value = "Value1")	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNoError.			
4	IF verify NOT OK THEN TP failed ENDIF			
5	**GetParameter** (Instance = instance1, Parameter = "Parameter1")	AppEm	⇒	
6		AppEm	⇐	**R: String**

7	IF (**String** equals to Value1)
	THEN TP passed
	ELSE TP failed
	ENDIF			

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_05	**Verify that parameter is set by the Front End Application for multiple instances (different parameter names).**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2.1			
Initial condition	Valid Instances, instance1 and instance2, are created.			
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**SetParameter** (Instance = instance1, Parameter = “Parameter1”, Value = “Value1”)	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNoError.			
4	IF verify NOT OK THEN TP failed ENDIF			
5	**GetParameter** (Instance = instance1, Parameter = “Parameter1”)	AppEm	⇒	
6		AppEm	⇐	**R: String**

7	IF (**String** equals to Value1)
	THEN GOTO step 8
	ELSE TP failed
	ENDIF			

8	**SetParameter** (Instance = instance2, Parameter = “Parameter2”, Value = “Value2”)	AppEm	⇒	
9		AppEm	⇐	**R: CEN175752Error**
10	Verify whether **CEN175752Error** equals to ERNoError.			
11	IF verify NOT OK THEN TP failed ENDIF			
12	**GetParameter** (Instance = instance2, Parameter = “Parameter2”)	AppEm	⇒	
13		AppEm	⇐	**R: String**

14	IF (**String** equals to Value2)
	THEN TP passed
	ELSE TP failed
	ENDIF			

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_06	**Verify that parameter is set by the Front End Application for multiple instances (the same parameter names).**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2.1			
Initial condition	Valid Instances, instance1 and instance2, are created.			
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**SetParameter** (Instance = instance1, Parameter = "Parameter1", Value = "Value1")	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNoError.			
4	IF verify NOT OK THEN TP failed ENDIF			
5	**GetParameter** (Instance = instance1, Parameter = "Parameter1")	AppEm	⇒	
6		AppEm	⇐	**R: String**

7	IF (**String** equals to Value1)
	THEN GOTO step 8
	ELSE TP failed
	ENDIF			

8	**SetParameter** (Instance = instance2, Parameter = "Parameter1", Value = "Value2")	AppEm	⇒	
9		AppEm	⇐	**R: CEN175752Error**
10	Verify whether **CEN175752Error** equals to ERNoError.			
11	IF verify NOT OK THEN TP failed ENDIF			
12	**GetParameter** (Instance = instance2, Parameter = "Parameter1")	AppEm	⇒	
13		AppEm	⇐	**R: String**

14	IF (**String** equals to Value2)
	THEN TP passed
	ELSE TP failed
	ENDIF			

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_07	**Verify that parameter is updated by the Front End Application.**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2.1			
Initial condition	A valid Instance, instance1, is created.			
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**SetParameter** (Instance = instance1, Parameter = “Parameter1”, Value = “Value1”)	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNoError.			
4	IF verify NOT OK THEN TP failed ENDIF			
5	**GetParameter** (Instance = instance1, Parameter = “Parameter1”)	AppEm	⇒	
6		AppEm	⇐	**R: String**

7	IF (**String** equals to Value1)
	THEN GOTO step 8
	ELSE TP failed
	ENDIF			

8	**SetParameter** (Instance = instance1, Parameter = “Parameter1”, Value = “Value2”)	AppEm	⇒	
9		AppEm	⇐	**R: CEN175752Error**
10	Verify whether **CEN175752Error** equals to ERNoError.			
11	IF verify NOT OK THEN TP failed ENDIF			
12	**GetParameter** (Instance = instance1, Parameter = “Parameter1”)	AppEm	⇒	
13		AppEm	⇐	**R: String**

14	IF (**String** equals to Value2)
	THEN GOTO step 8
	ELSE TP failed
	ENDIF			

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_08	**Verify that parameter's value is fetched by the Front End Application.**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2.1			
Initial condition	A valid Instance, instance1, is created. Parameter1 has already been set with value1.			
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**GetParameter** (Instance = instance1, Parameter = "Parameter1")	AppEm	⇒	
2		AppEm	⇐	**R: String**

3	IF (**String** equals to Value1)
	THEN TP passed
	ELSE TP failed
	ENDIF			

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_09	**Verify that parameter's value is fetched by the Front End Application (multiple instances).**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.2.1
Initial condition	Valid Instances, instance1, instance2 and instance3, are created. Parameter1 in instance1 has already been set with value1. Parameter2 in instance1 has already been set with value2. Parameter1 in instance3 has already been set with value3.

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**GetParameter** (Instance = instance1, Parameter = “Parameter1”)	AppEm	⇒	
2		AppEm	⇐	**R: String**

3	IF (**String** equals to Value1)
	THEN GOTO step 4
	ELSE TP failed
	ENDIF			

4	**GetParameter** (Instance = instance2, Parameter = “Parameter2”)	AppEm	⇒	
5		AppEm	⇐	**R: String**

6	IF (**String** equals to Value2)
	THEN GOTO step 7
	ELSE TP failed
	ENDIF			

7	**GetParameter** (Instance = instance3, Parameter = “Parameter1”)	AppEm	⇒	
8		AppEm	⇐	**R: String**

9	IF (**String** equals to Value3)
	THEN TP passed
	ELSE TP failed
	ENDIF			

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_10	**Verify that parameter is deleted by the Front End Application (single parameter).**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.2.1
Initial condition	A valid Instance, instance1, is created. Parameter1 has already been set by the Front End Application.

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**DeleteParameter** (Instance = instance1, Parameter = "Parameter1")	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNoError.			
4	IF verify NOT OK THEN TP failed ENDIF			
5	**GetParameter** (Instance = instance1, Parameter = "Parameter1")	AppEm	⇒	
6		AppEm	⇐	**R: String**

7	IF (**String** has invalid value)
	THEN TP passed
	ELSE TP failed
	ENDIF			

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_11	**Verify that parameter is deleted by the Front End Application (multiple parameters).**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.2.1
Initial condition	Valid Instances, instance1, instance2 and instance3, are created. Parameter1 in instance1 has already been set. Parameter2 in instance2 has already been set. Parameter1 in instance3 has already been set.

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**DeleteParameter** (Instance = instance1, Parameter = "Parameter1")	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNoError.			
4	IF verify NOT OK THEN TP failed ENDIF			
5	**GetParameter** (Instance = instance1, Parameter = "Parameter1")	AppEm	⇒	
6		AppEm	⇐	**R: String**

7	IF (**String** has invalid value)
	THEN GOTO step 8
	ELSE TP failed
	ENDIF			

8	**DeleteParameter** (Instance = instance2, Parameter = "Parameter2")		⇒	
9		AppEm	⇐	**R: CEN175752Error**
10	Verify whether **CEN175752Error** equals to ERNoError.	AppEm		
11	IF verify NOT OK THEN TP failed ENDIF			
12	**GetParameter** (Instance = instance2, Parameter = "Parameter2")		⇒	
13			⇐	**R: String**

14	IF (**String** has invalid value)
	THEN GOTO step 15
	ELSE TP failed
	ENDIF			

15	**DeleteParameter** (Instance = instance3, Parameter = "Parameter1")		⇒	
16			⇐	**R: CEN175752Error**

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

17	Verify whether **CEN175752Error** equals to ERNoError.			
18	IF verify NOT OK THEN TP failed ENDIF			
19	**GetParameter** (Instance = instance3, Parameter = “Parameter1”)		⇒	
20			⇐	**R: String**

21	IF (**String** has invalid value)
	THEN TP passed
	ELSE TP failed
	ENDIF			

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_12	**Verify whether StackAvail returns that communication stack is available.**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.5
Initial condition	A valid Instance, instance1, has already been created. Communication stack for instance1 is available.

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**StackAvail** (Instance = instance1)	AppEm	⇒	
2		AppEm	⇐	**R: Boolean**
3	IF returned TRUE THEN TP passed ELSE TP failed ENDIF			

TP_IH_API_BV_13	**Verify whether StackAvail returns that communication stack is available (multiple instances).**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.5
Initial condition	Valid Instances, instance1 and instance2, have already been created. Communication stack for instance1 and instance2 is available.

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**StackAvail** (Instance = instance1)	AppEm	⇒	
2		AppEm	⇐	**R: Boolean**

3	IF (returned FALSE)
	THEN TP failed
	ENDIF			

4	**StackAvail** (Instance = instance2)	AppEm	⇒	
5		AppEm	⇐	**R: Boolean**

6	IF (returned TRUE)
	THEN TP passed
	ELSE TP failed
	ENDIF			

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_14	**Verify whether StackAvail returns that communication stack is unavailable.**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.5
Initial condition	A valid Instance, instance1, has already been created. Communication stack for instance1 is not available.

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**StackAvail** (Instance = instance1)	AppEm	⇒	
2		AppEm	⇐	**R: Boolean**

3	IF (returned FALSE)
	THEN TP passed
	ELSE TP failed
	ENDIF			

TP_IH_API_BV_15	**Verify whether StackAvail returns that communication stack is unavailable (multiple instances).**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.5
Initial condition	Valid Instances, instance1 and instance2, have already been created. Communication stack for instance1 and instance2 are not available.

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**StackAvail** (Instance = instance1)	AppEm	⇒	
2		AppEm	⇐	**R: Boolean**

3	IF (returned TRUE)
	THEN TP failed
	ENDIF			

4	**StackAvail** (Instance = instance2)	AppEm	⇒	
5		AppEm	⇐	**R: Boolean**

6	IF (returned FALSE)
	THEN TP passed
	ELSE TP failed
	ENDIF			

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_16	**Verify whether StackAvail returns that communication stack is available (for first instance) and unavailable (for second instance).**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.5
Initial condition	Valid Instances, instance1 and instance2, have already been created. Communication stack for instance1 is available. Communication stack for instance2 is unavailable.

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**StackAvail** (Instance = instance1)	AppEm	⇒	
2		AppEm	⇐	**R: Boolean**

3	IF (returned FALSE)
	THEN TP failed
	ENDIF			

4	**StackAvail** (Instance = instance2)	AppEm	⇒	
5		AppEm	⇐	**R: Boolean**

6	IF (returned FALSE)
	THEN TP passed
	ELSE TP failed
	ENDIF			

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_17	**Dropping the instance with SENormal severity.**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2			
Initial condition	A valid Instance, instance1, has already been created. No session exists for instance1.			
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**DropInstance** (Instance = instance1, Severity = SENormal)	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNoError.			
4	IF verify OK THEN TP passed ELSE TP failed ENDIF			

TP_IH_API_BV_18	**Dropping the instance with SEUrgent severity.**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2			
Initial condition	A valid Instance, instance1, has already been created. No session exists for instance1.			
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**DropInstance** (Instance = instance1, Severity = SEUrgent)	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNoError.			
4	IF verify OK THEN TP passed ELSE TP failed ENDIF			

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_19	**Dropping the instance with SEUnconditional severity.**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2			
Initial condition	A valid Instance, instance1, has already been created. No session exists for instance1.			
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**DropInstance** (Instance = instance1, Severity = SEUnconditional)	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNoError.			
4	IF verify OK THEN TP passed ELSE TP failed ENDIF			

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_20	**Dropping the instance with SEUnconditional severity once session is in STStarting state.**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.2
Initial condition	A valid Instance, instance1, has already been created. Session for instance1 is being established. Session is in state STStarting. (NOTE See test purposes for other operations corresponding to this state, which specify the steps on how to fulfil this precondition.) Correct parametrization has already been done to establish session (example: IP address, port, URL, protocol, PDP context, etc. are set).

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**DropInstance** (Instance = instance1, Severity = SEUnconditional)	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNoError.			
4	IF verify OK THEN TP passed ELSE TP failed ENDIF			

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_21	**Dropping the instance with SEUnconditional severity once session is in STSessionIdle state.**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.2
Initial condition	A valid Instance, instance1, has already been created. Session for instance1 is established. Session is in state STSessionIdle. (NOTE See test purposes for other operations to this state, which specify the steps on how to fulfil this pre-condition.) Correct parametrization has already been done to establish session (example: IP address, port, URL, protocol, PDP context, etc. are set).

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**DropInstance** (Instance = instance1, Severity = SEUnconditional)	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNoError.			
4	IF verify OK THEN TP passed ELSE TP failed ENDIF			

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_22	**Dropping the instance with SEUnconditional severity once session is in STSendingADU state.**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.2
Initial condition	A valid Instance, instance1, has already been created. Session for instance1 is established. Session is in state STSendingADU. (NOTE See test purposes for other operations to this state, which specify the steps on how to fulfil this precondition.) Correct parametrization has already been done to establish session (example: IP address, port, URL, protocol, PDP context, etc. are set).

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**DropInstance** (Instance = instance1, Severity = S SEUnconditional)	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNoError.			
4	IF verify OK THEN TP passed ELSE TP failed ENDIF			

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_23	**Dropping the instance with SEUnconditional severity once session is in STSendingADURequest state.**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.2
Initial condition	A valid Instance, instance1, has already been created. Session for instance1 is established. Session is in state STSendingADURequest. (NOTE See test purposes for other operations to this state, which specify the steps on how to fulfil this precondition.) Correct parametrization has already been done to establish session (example: IP address, port, URL, protocol, PDP context, etc. are set).

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**DropInstance** (Instance = instance1, Severity = SEUnconditional)	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNoError.			
4	IF verify OK THEN TP passed ELSE TP failed ENDIF			

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_24	**Dropping the instance with SEUnconditional severity once session is in STSendingUnformattedADU state.**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.2
Initial condition	A valid Instance, instance1, has already been created. Session for instance1 is established. Session is in state STSendingUnformattedADU. (NOTE See test purposes for other operations to this state, which specify the steps on how to fulfil this precondition.) Correct parametrization has already been done to establish session (example: IP address, port, URL, protocol, PDP context, etc. are set).

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**DropInstance** (Instance = instance1, Severity = SEUnconditional)	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNoError.			
4	IF verify OK THEN TP passed ELSE TP failed ENDIF			

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_25	**Dropping the instance with SEUnconditional severity once session is in STSessionRxADUs state.**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.2
Initial condition	A valid Instance, instance1, has already been created. Session for instance1 is established. Session is in state STSessionRxADUs. (NOTE See test purposes for other operations to this state, which specify the steps on how to fulfil this pre-condition.) Correct parametrization has already been done to establish session (example: IP address, port, URL, protocol, PDP context, etc. are set).

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**DropInstance** (Instance = instance1, Severity = SEUnconditional)	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNoError.			
4	IF verify OK THEN TP passed ELSE TP failed ENDIF			

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_26	**Dropping the instance with SEUnconditional severity once session is in STErrored state.**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.2
Initial condition	A valid Instance, instance1, has already been created. Session for instance1 is established. Session is in state STErrored. (NOTE See test purposes for other operations to this state, which specify the steps on how to fulfil this precondition.) Correct parametrization has already been done to establish session (example: IP address, port, URL, protocol, PDP context, etc. are set).

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**DropInstance** (Instance = instance1, Severity = SEUnconditional)	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNoError.			
4	IF verify OK THEN TP passed ELSE TP failed ENDIF			

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BV_27	**Dropping the instance with SEUnconditional severity once session is in STEnding state.**
TP origin	Specific
Reference	ISO 17575-2:2016, 5.2
Initial condition	A valid Instance, instance1, has already been created. Session for instance1 is established. Session is in state STEnding. (NOTE See test purposes for other operations to this state, which specify the steps on how to fulfil this precondition.) Correct parametrization has already been done to establish session (example: IP address, port, URL, protocol, PDP context, etc. are set).

Stimulus and expected behaviour

	Tester	Test point		IUT
1	**DropInstance** (Instance = instance1, Severity = SEUnconditional)	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNoError.			
4	IF verify OK THEN TP passed ELSE TP failed ENDIF			

A.3.2 BI test purposes (Behaviour Invalid tests)

Test subgroup objective:

— to test IUT invalid behaviours with respect to instance initialization;

— to test IUT invalid behaviours with respect to parameter setting;

— to test IUT invalid behaviours with respect to parameter getting;

— to test IUT invalid behaviours with respect to parameter deleting;

— to test IUT invalid behaviours with respect to availability of communications stack;

— to test IUT invalid behaviours with respect to dropping the session including following severities:

  — SENormal;

  — SEUrgent;

in each state.

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BI_01	**Verify that FE Communications API returns invalid instance once FE Application selected invalid communication stack.**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2.1			
Initial condition	Set of Callback instances is instantiated.			
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**InitialiseInstance** (StackID = invalid-Stack, Callbacks = cb1)	AppEm	⇒	
2		AppEm	⇐	**R: Instance**
3	Verify whether **Instance** is invalid.			
4	IF verify OK THEN TP passed ELSE TP failed ENDIF			

TP_IH_API_BI_02	**Verify that FE Communications API returns invalid instance once FE Application provides invalid Callbacks.**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2.1			
Initial condition	FE Communications API shall handle at least one underlying communication stacks: which StackID equals to stack1.			
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**InitialiseInstance** (StackID = invalid-Stack, Callbacks = ø)	AppEm	⇒	
2		AppEm	⇐	**R: Instance**
3	Verify whether **Instance** is invalid.			
4	IF verify OK THEN TP passed ELSE TP failed ENDIF			

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BI_03	**Verify parameter setting upon invalid instance.**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2.1			
Initial condition				
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**SetParameter** (Instance = invalidInstance1, Parameter = “Parameter1”, Value = “Value1”) NOTE InvalidParameter1 may have too many characters, which are not handled by the FE Communications API.	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNotSet.			
4	IF verify OK THEN GOTO step 5 ELSE TP failed ENDIF			
5	**GetParameter** (Instance = instance1, Parameter = “InvalidParameter1”)	AppEm	⇒	
6		AppEm	⇐	**R: String**

7	IF (**String** is invalid)
	THEN TP passed
	ELSE TP failed
	ENDIF			

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BI_04	**Verify parameter setting upon invalid parameter.**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2.1			
Initial condition	A valid Instance, instance1, is created.			
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**SetParameter** (Instance = instance1, Parameter = "InvalidParameter1", Value = "Value1") NOTE InvalidParameter1 may have too many characters that are not handled by the FE Communications API.	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNotSet.			
4	IF verify NOT OK THEN TP failed ENDIF			
5	**GetParameter** (Instance = instance1, Parameter = "InvalidParameter1")	AppEm	⇒	
6		AppEm	⇐	**R: String**

7	IF (**String** is invalid)
	THEN TP passed
	ELSE TP failed
	ENDIF			

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BI_05	**Verify parameter setting upon invalid value.**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2.1			
Initial condition	A valid Instance, instance1, is created.			
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**SetParameter** (Instance = instance1, Parameter = “Parameter1”, Value = “InvalidValue1”) NOTE InvalidValue1 may have too many characters, which are not handled by the FE Communications API.	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNotSet.			
4	IF verify NOT OK THEN TP failed ENDIF			
5	**GetParameter** (Instance = instance1, Parameter = “Parameter1”)	AppEm	⇒	
6		AppEm	⇐	**R: String**

7	IF (**String** is invalid)
	THEN TP passed
	ELSE TP failed
	ENDIF			

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BI_06	**Verify getting the parameter's value upon invalid instance.**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2.1			
Initial condition				
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**GetParameter** (Instance = invalidInstance1, Parameter = “Parameter1”)	AppEm	⇒	
2		AppEm	⇐	**R: String**

3	IF (**String** is invalid)
	THEN TP passed
	ELSE TP failed
	ENDIF			

TP_IH_API_BI_07	**Verify getting the parameter's value once parameter does not exist.**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2.1			
Initial condition	A valid Instance, instance1, is created. InvalidParameter has not been set by the Front End Application.			
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**GetParameter** (Instance = instance1, Parameter = “invalidParameter”)	AppEm	⇒	
2		AppEm	⇐	**R: String**

3	IF (**String** is invalid)
	THEN TP passed
	ELSE TP failed
	ENDIF			

© ISO 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018

TP_IH_API_BI_08	**Verify parameter deletion upon invalid instance.**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2.1			
Initial condition				
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**DeleteParameter** (Instance = invalidInstance1, Parameter = “Parameter1”)	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNotSet.			
4	IF verify OK THEN TP passed ELSE TP failed ENDIF			

TP_IH_API_BI_09	**Verify parameter deletion once parameter does not exist.**			
TP origin	Specific			
Reference	ISO 17575-2:2016, 5.2.1			
Initial condition	A valid Instance, instance1, is created.			
Stimulus and expected behaviour				
	Tester	**Test point**		**IUT**
1	**DeleteParameter** (Instance = instance1, Parameter = “Parameter1”)	AppEm	⇒	
2		AppEm	⇐	**R: CEN175752Error**
3	Verify whether **CEN175752Error** equals to ERNotSet.			
4	IF verify OK THEN TP passed ELSE TP failed ENDIF			

© ISO 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD CEN ISO/TR 16401-1:2018