ETSI GS CIM 027 V1.1.1 (2024-04)

Context Information Management (CIM); NGSI-LD Testing Framework: Test Purposes Description Language (TPDL)

Disclaimer

The present document has been produced and approved by the cross-cutting Context Information Management (CIM) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
DGS/CIM-0027

Keywords
API, IoT, NGSI-LD, testing

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) cross-cutting Context Information Management (CIM).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

The present document defines a template for the Test Purposes (TP) description.

Introduction

The ISG CIM group has defined an API for exchange of information contextualized in time, space and relation to other information, using a property graph model, with the intent that the associated protocol (called NGSI-LD) becomes the "glue" between all kinds of applications and databases associated with services for Smart Cities, Smart Agriculture, Smart Manufacturing, etc.

To be successful, the NGSI-LD API specification needs to be well understood and well implemented. The community of users will not be solely highly professional engineers employed by big companies but will include many small teams and SMEs and even hobbyists. Therefore, it is essential that the developers have access to not only the standard but also a test specification and a testing environment to check that their work is (and remains) conformant to the ETSI NGSI-LD specification.

The developers will usually write integration tests to validate the behaviour of their NGSI-LD implementation, but it is important to assert compliance to the specification based on a test suite agreed by the group creating the API specification, i.e. ETSI ISG CIM. Therefore, it is very important to create a set of ETSI-approved test cases.

What is more, the existence of such a test suite will likely help to increase the adoption of the NGSI-LD specification by giving developers a ready to use and complete set of sample requests.

The present document defines a template for the Test Purposes (TP) description.

1 Scope

The present document defines a template for the Test Purposes (TP) description.

The Testing Framework documents specify a framework and methodology for the development of the test strategies, test systems and resulting test specifications. The present document identifies the implementation under test (scope of the testing), the format for the test specification, the test architecture, the points of control and observation, the naming conventions (e.g. for test case ID and test case grouping ID), etc.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI GS CIM 009 (V1.5.1): "Context Information Management (CIM); NGSI-LD API".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

Not applicable.

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

NOTE: The letters "NGSI-LD" were added to most terms to confirm that they are distinct from other terms of similar/same name in use in other organizations, however, in the present document the letters "NGSI-LD" are generally omitted for brevity.

NGSI-LD Central Broker: NGSI-LD Context Broker that only uses a local storage when serving NGSI-LD requests, without involving any external Context Sources

NGSI-LD Context Broker: architectural component that implements all the NGSI-LD interfaces

NGSI-LD Context Consumer: agent that uses the query and subscription functionality of NGSI-LD to retrieve context information

NGSI-LD Context Producer: agent that uses the NGSI-LD context provision and/or registration functionality to provide or announce the availability of its context information to an NGSI-LD Context Broker

NGSI-LD Context Registry: software functional element where Context Sources register the information that they can provide

NOTE: It is used by Distribution Brokers and Federation Brokers to find the appropriate Context Sources which can provide the information required for serving an NGSI-LD request.

NGSI-LD Context Source: source of context information which implements the NGSI-LD consumption and subscription (and possibly provision) interfaces defined by the present document

NOTE: It is usually registered with an NGSI-LD Registry so that it can announce what kind of information it can provide, when requested, to Context Consumers and Brokers.

NGSI-LD Distribution Broker: NGSI-LD Context Broker that uses both local context information and registration information from an NGSI-LD Context Registry, to access matching context information from a set of distributed Context Sources

NGSI-LD Federation Broker: Distribution Broker that federates information from multiple underlying NGSI-LD Context Brokers and across domains

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

API	Application Programming Interface
ATS	Abstract Test Suite
GS	Group Specification
ICS	Implementation Conformance Statement
IoT	Internet of Things
ISG	Industry Specification Group
NGSI	Next Generation Service Interfaces
PICS	Profile Implementation Conformance Statement
SUT	System Under Test
TP	Test Purposes
TSS	Test Suite Structure
TSS&TP	Test Suite Structure & Test Purposes
URL	Universal Resource Locator

4 Prerequisites and Test Configurations

4.1 Test Configurations

Test configurations are defined upon the different architectures' options defined in clause 4.3 of ETSI GS CIM 009 [1]. Considered architectures are:

- **Centralized architecture:** A Central Broker stores all the context information. There are Context Producers that use update operations to update the context information in the Central Broker and there are Context Consumers that request context information from the Central Broker, either using synchronous one-time query or asynchronous subscribe/notify operations.

- **Distributed architecture:** All information is stored by the Context Sources. Context Sources implement the query and subscription part of the NGSI-LD API as a Context Broker does. They register themselves with the Context Registry, providing information about what context information they can provide, but not the context information itself.

- **Federated architecture:** The architecture works in the same way as the distributed architecture described in clause 4.3.3 of ETSI GS CIM 009 [1], except that instead of simple Context Sources, whole domains are registered with the respective Context Broker as point of access. Typically, the domains will be registered to the federation Context Registry on a more coarse-grained level, providing scopes, in particular geographic scopes, that can then be matched to the scopes provided in the requests.

Test configurations are defined to test different entities such as NGSI-LD Broker, NGSI-LD Context Producer, NGSI-LD Context Consumer, NGSI-LD Context Source, etc.

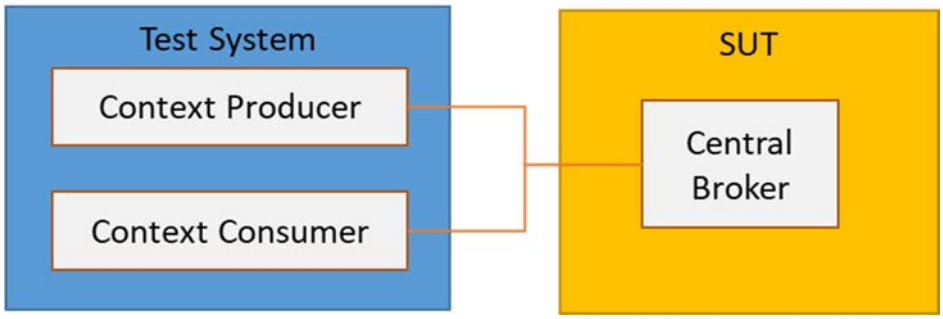

Figure 4.1-1: Test configuration 1 (CF_01)

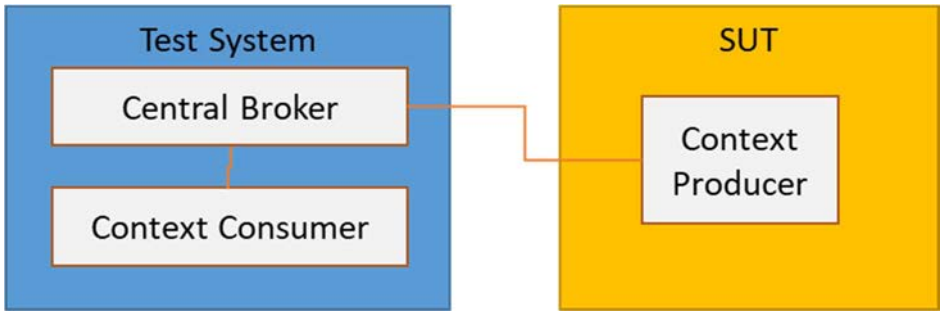

Figure 4.1-2: Test configuration 2 (CF_02)

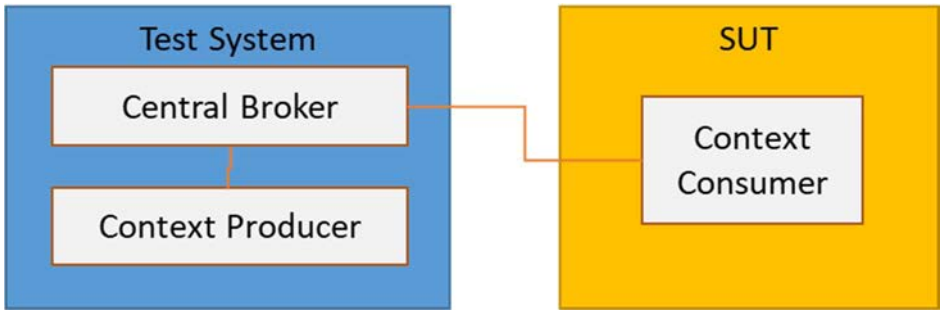

Figure 4.1-3: Test configuration 3 (CF_03)

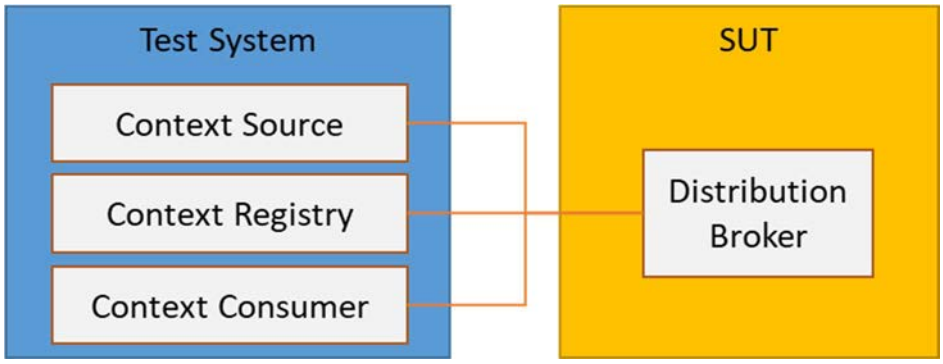

Figure 4.1-4: Test configuration 4 (CF_04)

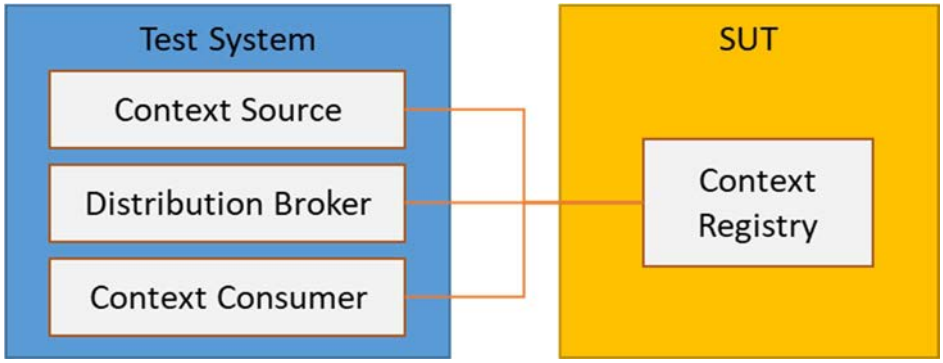

Figure 4.1-5: Test configuration 5 (CF_05)

5 Test Purposes (TP)

5.1 Introduction

A test purpose is an informal description of the expected test behaviour. As such it is written in prose.

When needed to clarify the TP, it is helpful to add some graphical presentations, mainly tables, and include message sequence charts.

5.2 TP definition conventions

In order to increase the readability of the TP, the following two recommendations should be followed:

- Each TP should be presented in a table, containing two main parts:
  - The TP header, which contains the TP identifier, the TP objective and the external references (ICS and base standard).
  - The behaviour part, which contains the test behaviour description. This part can be optionally divided in the three following parts, in order to increase the readability:
    - the initial conditions;
    - the expected behaviour;
    - the final conditions.
- The prose describing the test behaviour (including initial and final conditions) should follow some rules, as for instance the use of reserved keywords and syntax.

Table 5.2-1: TP template

TP Id		
Test objective		
References		
Config Id		
Parent Release		
PICS Selection		
Initial conditions		
Permutations		
TP Permutation Id		
Name		
Description		
Expected behaviour	**Test events**	**Direction**
	when { **}**	SUT ← Client
	then { **}**	SUT → Client

Table 5.2-2: Description of the fields of the TP template

TP Header	
TP ID	The TP ID is a unique identifier. It shall be specified according to the TP naming conventions defined in clause 5.3.
Test objective	Short description of test purpose objective according to the requirements from the base standard. The test objective clearly indicates which requirement is intended to be tested in the test purpose. This part eases the understanding of the TP behaviour. This also eases the identification of the requirements, which were used as a basis for the test purpose. It is recommended to limit the length of the test objective to one sentence.
References	The reference indicates the clauses of the reference standard specifications in which the conformance requirement is expressed. In the reference row, the TP writer indicates, in which clauses of the protocol standards, the requirement is expressed. This information is critical, because it justifies the existence and the behaviour of the TP. The reference row may refer to several clauses. When the clause containing the requirement is big (for instance, more than ½ page), it is recommended to indicate the paragraph of the clause where the requirement was identified. The reference to the base standard actually is precise enough to enable the TP reader to identify quickly and precisely the requirement.
Config ID	The Config ID is a unique identifier for the configuration required to execute the test, as detailed in clause 4.1.
Parent Release	The parent release identifies the version of the reference standard specification in which the requirement has been introduced.
PICS selection	The PICS selection row contains a Boolean expression, made of ICS parameters. It is recommended to use ICS acronym, which clearly identify the role of the ICS. A mapping table is included in the TP document to link the ICS acronym with its corresponding reference in the ICS document.
Initial conditions	The initial conditions define in which initial state the SUT has to be to apply the actual TP. In the corresponding Test Case, when the execution of the initial condition does not succeed, it leads to the assignment of an inconclusive verdict.
Permutations	The permutations section introduces the list of permutations that are part of the current TP.
TP Permutation Id	The TP Permutation Id is a unique identifier. It shall be specified according to the TP naming conventions defined in clause 5.3.
Name	The name is a unique and short name identifying the permutation. It is a composed of the final part of the TP permutation Id and a key describing the purpose of the test.
Description	Short description of the TP permutation. It describes the specific behaviour that is tested by the permutation.
Expected behaviour	Definition of the events, which are parts of the TP objective, and the SUT are expected to perform in order to conform to the base specification. In the corresponding Test Case, Pass or Fail verdicts can be assigned there.

5.3 TP Identifier naming conventions

The TP identifier identifies uniquely the test purposes. In order to ensure the uniqueness of the TP identifier, it follows a naming convention.

The more useful and straightforward naming convention consists of using the test suite structure, to form the first part of the TP identifier. Then the final part consists of a number to identify the TP order within a TP group.

The TP identifier is formed by the abbreviation "TP", followed by abbreviation representing the group of the following TSS levels. Each field of the TP identifier is separated by a "/".

Thus, a TP shall be named following this convention:

TP/NGSI-LD/<Group>/<Sub Group>/<Sub Sub Group>/<Test Scenario>_<Variant>_<Permutation_Index>

5.4 Rules for the behaviour description

The following global rules apply, when writing the behaviour description:

- The behaviour description is written in an explicit, exhaustive and unambiguous manner.
- The behaviour description only refers to externally observable test events (send/receive PDUs, timer, counters, etc.) or to events or states, which can be directly or indirectly observed externally.
- All test events used in the behaviour description are part of the procedures specified in the standards.
- The wording of the test events in the behaviour description is explicit, so that the ATS writers do not have to interpret the behaviour description.
- All test events in the behaviour description should result as far as possible in one ATS statement.

The test behaviour is described in prose. This enables to use different ways to express similar behaviour. But using different expressions to define identical behaviours can lead to some misinterpretation of the test purposes. Also, the meaning and the expected order of the test events have a clear and unique meaning for different readers.

Thus, the present document recommends to use pre-defined keywords in order to express clearly and uniquely the test behaviour.

Table 5.4-1 shows some recommended pre-defined keywords and their context of usage. The pre-defined keywords are also likely to be used in combination with the "{" "}"delimiters, in order to clearly delimitate their action in the test behaviour description.

Table 5.4-1 does not present an exhaustive list, so that additional keywords might be defined as necessary. The definition of additional keywords is included in the corresponding TSS&TP document.

Table 5.4-1: List of pre-defined keywords for the behaviour description

Behavioural keywords	
with	with, together with "{" "}" delimiters is used to express the initial conditions, which consist of a set of events, to be executed before starting with the test behaviour corresponding to the test objective. EXAMPLE 1: with { the SUT having sent a container create request message and ... }

ensure that	ensure that, together with "{" "}" delimiters is used to define the place of the expected behaviour (TP body) or the final conditions.
	EXAMPLE 2:
	ensure that {
	when { the SUT receives a valid container create request message... }
	}

when/then	when combined with then enables to define the test behaviour involving a combination of stimuli and response events. The when/then combination is used when the occurrence of an event is triggered by the realization of a previous event.
	EXAMPLE 3:
	ensure that {
	when {
	a XXX signal is activated }
	then {
	the SUT sends a message containing YYY Value indicating "True"
	}
	}

Event keywords	
the SUT	event in the TP is expressed from the point of view of the SUT. This avoids any misinterpretation.
receives	states for an event corresponding to the receipt of a message by the SUT.
having received	states for a condition where the SUT has received a message.
sends	states for an event corresponding to the sending of a message by the SUT.
having sent	states for a condition where the SUT has sent a message.

from/to	indicates the destination or the origin of a message as necessary (interface, ...)
	EXAMPLE 4:
	ensure that {
	when { the SUT receives a valid XXX message from the YYY port.. }
	}

on expiry of	indicates the expiry of a timer, being a stimulus for forthcoming event. EXAMPLE 5: ensure that { on expiry of the Timer T1, the SUT sends a valid XXX message... }
after expiry of	used to indicate that an event is expected to occur after the expiry of a timer. EXAMPLE 6: ensure that { the SUT sends a valid XXX message after expiry of the minimum timer interval }
before expiry of	used to indicate that an event is expected to occur before the expiry of a timer. EXAMPLE 7: ensure that { the SUT sends a valid XXX message before expiry of the maximum timer interval }
Event attribute keywords	
valid	indicates that the event sent or received is a valid message according to the protocol standard, thus: • containing all mandatory parameters, with valid field values; • containing required optional fields according to the protocol context, with valid field values.
invalid	indicates that the event sent or received is an invalid message according to the protocol standard. Further details describing the invalid fields of the message is added. EXAMPLE 8: with { the SUT having sent an invalid XXX message containing no mandatory YYY parameter... }
containing	enables to describe the content of a sent or received message.
indicating	enables to specify the interpretation of the value allocated to a message parameter. EXAMPLE 9: with { the SUT having sent a valid XXX message containing a mandatory YYY parameter indicating "ZZZ supported"... }
Logical keywords	
and	used to combine statements of the behaviour description.
or	
not	

Create Entity TP example:

TP Id	TP/NGSI-LD/CI/Prov/E/001_01	
Test objective	Check that the SUT accepts an Entity create request	
Reference	ETSI GS CIM 009 (V1.5.1) [1], clause 5.6.1	
Config Id	CF_01	
Parent Release	V1.3.1	
PICS Selection	PICS_EntCre	

Initial conditions	with {
	the SUT being in the "initial state"
	}	

Permutations		
TP Permutation Id	TP/NGSI-LD/CI/Prov/E/001_01_01	
Name	001_01_01 MinimalEntity	
Description	Check that a minimal entity can be created	
Expected behaviour	**Test events**	**Direction**
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/' method set to 'POST' Request Header['Content-Type'] set to 'application/json' and payload defined in file: 'building-minimal-without-context-sample.jsonld' }	SUT ← Client

	then {
	the SUT sends a valid Response for the operations:
	Create Entity Selecting Content Type with Response Status Code set to 201 and
	Create Entity Selecting Content Type with Response Header: Location containing $${entity_id} and
	Retrieve Entity by Id with Check Created Resource Set To and
	Query Parameter: 'created_resource' set to 'created_entity' and
	Query Parameter: 'response_body' set to 'response1.json()' and
	Query Parameter: 'ignored_keys' set to '${None}'
	}
		SUT → Client

Annex A (informative): Change history

Date	Version	Information about changes
May, 30th 2023	0.0.1	Early draft of the document corresponding to the TTF2 activity.
September, 30th 2023	0.0.2	Stable draft of the document corresponding to the TTF2 activity.
March 2024	0.0.3	Final draft for approval.
March 2024	0.0.4	Typos, formatting. ETSI Technical Officer review after TB approval for ETSI EditHelp publication pre-processing.
April 2024	1.1.1	First published version

History

Document history		
V1.1.1	April 2024	Publication