ETSI TS 103 597-1 V1.1.1 (2021-01)

TECHNICAL SPECIFICATION

Methods for Testing and Specification (MTS); Test Specification for MQTT; Part 1: Conformance Tests

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 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

Contents

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ETSI TS 103 597-1 V1.1.1 (2021-01)
https://standards.iteh.ai/catalog/standards/sist/2f395ea-19c0-4cbd-a67b-
d7e0ae4d3239/etsi-ts-103-597-1-v1-1-1-2021-01

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is 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 IPR Policy, no investigation, 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.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Methods for Testing and Specification (MTS).

The present document is part 1 of a multi-part deliverable covering the MQTT protocol as identified below:

Part 1: **"Conformance Tests"**;

Part 2: "Security Tests";

Part 3: "Performance Tests".

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.

Introduction

While the Internet of Things (IoT) is on the rise, the quality assurance of interconnected systems becomes an ever-increasing challenge. Within the last years, many different IoT protocols came to the fore. The MQ Telemetry Transport (MQTT) protocol is one of the most popular representatives as many surveys have shown.

Although many implementations for the MQTT protocol exist, it lacks in satisfying quality assurance. While many IoT components communicate over standardized protocols, communication protocols for IoT like MQTT or CoAP evolved over time without a holistic approach for quality assurance.

In the present document the conformance testing is presented. It provides a basis for interoperability testing and performance testing. The latter is presented in ETSI TS 103 597-3 [i.3].

1 Scope

The present document provides a test specification, i.e. an overall test suite structure and catalogue of test purposes for the MQ Telemetry Transport (MQTT). It will be a reference base for both client-side test campaigns and server-side test campaigns addressing the conformance issues.

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] OASIS Standard: "MQTT Version 3.1.1".

[2] ETSI ES 203 119-4: "Methods for Testing and Specification (MTS); The Test Description Language (TDL); Part 4: Structured Test Objective Specification (Extension)".

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.

[i.1] ISO/IEC 9646-1: " Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework -- Part 1: General concepts".

[i.2] ETSI ES 202 951: "Methods for Testing and Specification (MTS); Model-Based Testing (MBT); Requirements for Modelling Notations".

[i.3] ETSI TS 103 597-3: "Methods for Testing and Specification (MTS); Test Specification for MQTT; Part 3: Performance Tests".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

conformance: extent to which an implementation of a standard satisfies the requirements expressed in that standard

conformance testing: process to verify to what extent the IUT conforms to the standard

Implementation Under Test (IUT): implementation of one or more Open Systems Interconnection (OSI) protocols in an adjacent user/provider relationship, being the part of a real open system, which is to be studied by testing (ISO/IEC 9646-1 [i.1])

system under test: real open system in which the implementation under test resides (ETSI ES 202 951 [i.2])

test purpose: non-formal high-level description of a test, mainly using text

test suite structure: document defining (hierarchical) grouping of test cases according to some rules

3.2 Symbols

Void.

3.3 Abbreviations

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

IUT	Implementation Under Test
MQTT	MQ Telemetry Transport
SUT	System Under Test
TDL	Test Description Language
TDL-TO	Test Description Language - Test Objectives
TSS	Test Suite Structure

4 Test Suite Structure

4.0 Introduction

In the first one an MQTT server as SUT is considered and in the latter, an MQTT client as SUT is considered

The structure itself is partly derived from the MQTT specification [1] but changed due to overlapping functions that cannot be tested separately.

4.1 Broker as SUT

1) All mandatory message data fields

    a) CONNECT Control Packet

        i) Fixed Header

            1) Header Flags

        ii) Variable Header

            1) Protocol Name

            2) Protocol Level

            3) Reserved Flags

            4) Last Will Testament Flags

            5) Credentials Flags

    iii) Payload

        1)   Client Identifier

        2)   Will Topic

        3)   Credentials

b)   CONNACK Control Packet

    i)   Fixed Header

    ii) Variable Header

        1)   Clean Session

        2)   Present Session

        3)   Return Codes

c)   SUBSCRIBE Control Packet

    i)   Fixed Header

        1)   Header Flags

    ii) Variable Header

        1)   Packet Identifier

    iii) Payload

        1)   UTF-8 Encoding

        2)   Topic Filter

        3)   Requested QoS

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ETSI TS 103 597-1 V1.1.1 (2021-01)
https://standards.iteh.ai/catalog/standards/sist/2f3395ea-19c0-4cbd-a67b-
d7e0ae4d3239/etsi-ts-103-597-1-v1-1-1-2021-01

d)   SUBACK Control Packet

    i)   Fixed Header

        1)   Header Flags

    ii) Variable Header

        1)   Packet Identifier

    iii) Payload

        1)   Return Codes

e)   UNSUBSCRIBE Control Packet

    i)   Fixed Header

        1)   Header Flags

    ii) Variable Header

        1)   Packet Identifier

    iii) Payload

        1)   UTF-8 Encoding

        2)   Topic Filters

    f)   UNSUBACK Control Packet

        i)   Fixed Header

        ii)   Variable Header

    g)   PINGREQ Control Packet

        i)   Fixed Header

    h)   PINGRESP Control Packet

        i)   Fixed Header

    i)   DISCONNECT Control Packet

        i)   Fixed Header

2)   Protocol features

    a)   General

        i)   QoS levels

        ii)   Delivery retransmission

        iii)   Retained messages

        iv)   Message ordering

        v)   Anonymous client identifier

    b)   Connect/disconnect (session handling)

        i)   Credentials

        ii)   Session initiation

        iii)   Session states

    c)   Subscribe

    d)   Unsubscribe

    e)   Immediate publish (w/o awaiting for CONNACK)

    f)   Last Will and Testament message

    g)   Heartbeats: keepAlive values

    h)   Topic names/filters

    i)   Error handling

4.2   Client as SUT

1)   All mandatory message data fields

    a)   CONNECT Control Packet

    b)   CONNACK Control Packet

    c)   PUBLISH Control Packet

    d)   PUBACK Control Packet

    e)   PUBREC Control Packet

    f)   UNSUBACK Control Packet

    g)   PUBREL Control Packet

    h)   PUBCOMP Control Packet

    i)   SUBSCRIBE Control Packet

    j)   UNSUBSCRIBE Control Packet

    k)   DISCONNECT Control Packet

2)   Protocol features

    a)   keepAlive values

4.3　　TP naming convention

TPs are numbered, starting at 001, within each main scope. The main scopes are organized according to the TSS. Some TPs may not have a second level scope.

Table 1: TP identifier naming convention scheme

Identifier: **TP_<protocol>_<iut>_<scope>_<2nd_lvl_scope>*_<number>***				
TP	=	Test Purpose	Fixed to TP	
<protocol>	=	Protocol name	Fixed to MQTT	
<iut>	=	Type of IUT	Client or Broker	
<scope>	=	Main scope	Scope of the protocol (feature)	
			<CONTROL PACKET>	Name of the scoped Control Packet
			FEAT	Protocol Features
<2nd_lvl_scope>	=	Second level scope	RTND	Retained Messages
<number>	=	Sequential number	From 001 to 999	
*optional				

4.4　　TP structure

Each TP has been written in TDL-TO and thus in a structured manner which is consistent with all other TPs. The intention of this is to make the TPs more formal. In addition, a more readable format is provided by generating tables out of the TDL-TO format. The defined structure, that has been used, is illustrated in table 2. This table should be read in conjunction with any TP, i.e. please use a TP as an example to facilitate the full comprehension of table 2. All structures are defined formally in the TDL Specification ETSI ES 203 119-4 [2].

Table 2: Structure of a single TP

TP part	Text	Example
Header	<Identifier> <Test objective> <Reference> <PICS reference>	*see table 1* *"The IUT has to close network connect …"* *[MQTT-3.2.2-6]* *PIC_BROKER_BASIC*
Initial condition (optional)	Free text description of the condition that the IUT has reached before the test purpose applies.	*… the IUT entity having a present session for the CLIENT_ID entity …*
Start point	Describes the full logic of the test purpose. Includes trigger and expected behavior of the IUT.	*Expected behavior* *ensure that { … }*
Trigger	One or more actions that trigger an expected response of the IUT. Mostly a set of different messages the IUT receives.	*when {* * the IUT entity receives a CONNECT message containing header_flags indicating value '1111'B;* *}*
Expected behavior	Describes the response that the IUT sends after receiving a certain (set of) messages. This response describes the pass criteria	*then {* * the IUT entity closes the TCP_CONNECTION* *}*

5   Test Purposes for MQTT Broker

TP Id	TP_MQTT_BROKER_CONNECT_001
Test Objective	Verify that the IUT closes the network connection if fixed header flags in CONNECT Control Packet are invalid.
Reference	[MQTT-2.2.2-1], [MQTT-2.2.2-2], [MQTT-3.1.4-1], [MQTT-3.2.2-6]
PICS Selection	PICS_BROKER_BASIC
Initial Conditions	
Expected Behaviour	

ensure that {
  when {
    the IUT receives a CONNECT message containing
    header_flags indicating value '1111'B;
  }
  then {
    the IUT closes the TCP_CONNECTION
  }
}	

Final Conditions	

TP Id	TP_MQTT_BROKER_CONNECT_002
Test Objective	Verify that the IUT either disconnects the client or continues processing the CONNECT Control Packet if the protocol name does not correspond to 'MQTT'.
Reference	[MQTT-3.1.2-1], [MQTT-3.1.4-4]
PICS Selection	PICS_BROKER_BASIC
Initial Conditions	
Expected Behaviour	

```
ensure that {
  when {
      the IUT receives a CONNECT message containing
      header_flags indicating value '0000'B,
      protocol_name corresponding to PROTOCOL_NAME_INVALID,
      protocol_level indicating value 0x04;
  }
  then {
      the IUT closes the TCP_CONNECTION
      // TODO: missing in TTCN-3 Implementation
      or the IUT sends a CONNACK message containing
      connect_return_code indicating value 0x00;
  }
}
```

Final Conditions	

TP Id	TP_MQTT_BROKER_CONNECT_003
Test Objective	Verify that the IUT responds to supported protocol levels (in scope: MQTT-3.1.1) with the return code 0x00.
Reference	[MQTT-3.1.2-2], [MQTT-3.1.4-4]
PICS Selection	PICS_BROKER_BASIC
Initial Conditions	
Expected Behaviour	

```
ensure that {
  when {
      the IUT receives a CONNECT message containing
      header_flags indicating value '0000'B,
      protocol_name corresponding to PROTOCOL_NAME,
      protocol_level indicating value 0x04;
  }
  then {
      the IUT sends a CONNACK message containing
      connect_return_code indicating value 0x00;
  }
}
```

Final Conditions	

TP Id	TP_MQTT_BROKER_CONNECT_004
Test Objective	Verify that the IUT validates the reserved flags in the CONNECT Control Packet.
Reference	[MQTT-3.1.2-3], [MQTT-3.1.4-1], [MQTT-3.2.2-6]
PICS Selection	PICS_BROKER_BASIC
Initial Conditions	
Expected Behaviour	

```
ensure that {
    when {
        the IUT receives a CONNECT message containing
        header_flags indicating value '0000'B,
        protocol_name corresponding to PROTOCOL_NAME,
        protocol_level indicating value 0x04,
        connect_flags containing
            reserved_field indicating value '1'B;
        ;
    }
    then {
        the IUT closes the TCP_CONNECTION
    }
}
```

Final Conditions	

TP Id	TP_MQTT_BROKER_CONNECT_005
Test Objective	Verify that the IUT validates the will_topic and will_message fields if the will_flag is set to 1.
Reference	[MQTT-3.1.2-9], [MQTT-3.1.4-1], [MQTT-3.2.2-6]
PICS Selection	PICS_BROKER_BASIC and PICS_BROKER_LWT
Initial Conditions	
Expected Behaviour	

```
ensure that {
    when {
        the IUT receives a CONNECT message containing
        header_flags indicating value '0000'B,
        protocol_name corresponding to PROTOCOL_NAME,
        protocol_level indicating value 0x04,
        connect_flags containing
            will_flag indicating value '1'B,
            will_qos corresponding to AT_MOST_ONCE,
            will_retain indicating value '0'B,
            reserved_field indicating value '0'B;
        ,
        payload containing
            will_topic indicating value omit,
            will_message indicating value omit;
        ;
    }
    then {
        the IUT closes the TCP_CONNECTION
    }
}
```

Final Conditions	

TP Id	TP_MQTT_BROKER_CONNECT_006
Test Objective	Verify that the IUT validates the the will_topic and will_message fields to be omitted if the will_flag is set to 0.
Reference	[MQTT-3.1.2-11], [MQTT-3.1.4-1], [MQTT-3.2.2-6]
PICS Selection	PICS_BROKER_BASIC and PICS_BROKER_LWT and PICS_BROKER_RTND
Initial Conditions	
Expected Behaviour	

```
ensure that {
   when {
       the IUT receives a CONNECT message containing
       header_flags indicating value '0000'B,
       protocol_name corresponding to PROTOCOL_NAME,
       protocol_level indicating value 0x04,
       connect_flags containing
          will_flag indicating value '0'B,
          will_qos corresponding to AT_LEAST_ONCE,
          will_retain indicating value '1'B,
          reserved_field indicating value '0'B;

       ,
       payload containing
          will_topic corresponding to PX_WILL_TOPIC,
          will_message corresponding to PX_WILL_MESSAGE;
       ;
   }
   then {
       the IUT closes the TCP_CONNECTION
   }
}
```

Final Conditions

TP Id	TP_MQTT_BROKER_CONNECT_007
Test Objective	Verify that the IUT validates the will_qos field to be set to 0 if the will_flag is set to 0.
Reference	[MQTT-3.1.2-13], [MQTT-3.1.4-1], [MQTT-3.2.2-6]
PICS Selection	PICS_BROKER_BASIC
Initial Conditions	
Expected Behaviour	

```
ensure that {
   when {
       the IUT receives a CONNECT message containing
       header_flags indicating value '0000'B,
       protocol_name corresponding to PROTOCOL_NAME,
       protocol_level indicating value 0x04,
       connect_flags containing
          will_flag indicating value '0'B,
          will_qos corresponding to AT_LEAST_ONCE,
          will_retain indicating value '0'B,
          reserved_field indicating value '0'B;
       ;
   }
   then {
       the IUT closes the TCP_CONNECTION
   }
}
```

Final Conditions

TP Id	TP_MQTT_BROKER_CONNECT_008
Test Objective	Verify that the IUT validates the will_qos field and rejects connections with an invalid will_qos value.
Reference	[MQTT-3.1.2-14], [MQTT-3.1.4-1], [MQTT-3.2.2-6]
PICS Selection	PICS_BROKER_BASIC and PICS_BROKER_LWT
Initial Conditions	
Expected Behaviour	

ensure that {
   when {
       the IUT receives a CONNECT message containing
       header_flags indicating value '0000'B,
       protocol_name corresponding to PROTOCOL_NAME,
       protocol_level indicating value 0x04,
       connect_flags containing
         will_flag indicating value '1'B,
         will_qos corresponding to INVALID_QOS,
         will_retain indicating value '0'B,
         reserved_field indicating value '0'B;
       ;
   }
   then {
       the IUT closes the TCP_CONNECTION
   }
 }	

Final Conditions	

TP Id	TP_MQTT_BROKER_CONNECT_009
Test Objective	Verify that the IUT validates the will_qos field if the will flag is set to 1.
Reference	[MQTT-3.1.2-14], [MQTT-3.1.4-4]
PICS Selection	PICS_BROKER_BASIC and PICS_BROKER_LWT
Initial Conditions	
Expected Behaviour	

ensure that {
   when {
       the IUT receives a CONNECT message containing
       header_flags indicating value '0000'B,
       protocol_name corresponding to PROTOCOL_NAME,
       protocol_level indicating value 0x04,
       connect_flags containing
         will_flag indicating value '1'B,
         will_qos corresponding to AT_MOST_ONCE,
         will_retain indicating value '0'B,
         reserved_field indicating value '0'B;
       ;
   }
   then {
       the IUT sends a CONNACK message containing
       connect_return_code indicating value 0x00;
   }
 }	

Final Conditions	

TP Id	TP_MQTT_BROKER_CONNECT_010
Test Objective	Verify that the IUT validates the will_flag and will_retain flags to be set correctly.
Reference	[MQTT-3.1.2-15], [MQTT-3.1.4-1], [MQTT-3.2.2-6]
PICS Selection	PICS_BROKER_BASIC
Initial Conditions	
Expected Behaviour	

```
ensure that {
    when {
        the IUT receives a CONNECT message containing
        header_flags indicating value '0000'B,
        protocol_name corresponding to PROTOCOL_NAME,
        protocol_level indicating value 0x04,
        connect_flags containing
            will_flag indicating value '0'B,
            will_qos corresponding to AT_MOST_ONCE,
            will_retain indicating value '1'B,
            reserved_field indicating value '0'B;
        ,
        payload containing
            will_topic indicating value omit,
            will_message indicating value omit;
        ;
    }
    then {
        the IUT closes the TCP_CONNECTION
    }
}
```

| **Final Conditions** | |
| | |

TP Id	TP_MQTT_BROKER_CONNECT_011
Test Objective	Verify that the IUT validates the will_retain flag to be set to 0 if the will_flag is set to 0.
Reference	[MQTT-3.1.2-15], [MQTT-3.1.4-4]
PICS Selection	PICS_BROKER_BASIC
Initial Conditions	
Expected Behaviour	

```
ensure that {
    when {
        the IUT receives a CONNECT message containing
        header_flags indicating value '0000'B,
        protocol_name corresponding to PROTOCOL_NAME,
        protocol_level indicating value 0x04,
        connect_flags containing
            will_flag indicating value '0'B,
            will_qos corresponding to AT_MOST_ONCE,
            will_retain indicating value '0'B,
            reserved_field indicating value '0'B;
        ,
        payload containing
            will_topic indicating value omit,
            will_message indicating value omit;
        ;
    }
    then {
        the IUT sends a CONNACK message containing
        connect_return_code indicating value 0x00;
    }
}
```

| **Final Conditions** | |
| | |