

Test suite: MQTT Conformance

TestSuite

Test Case	Short Description	TTCN-3
tc_mqtt_broker_connect_001	The IUT MUST close the network connection if fixed header flags in CONNECT Control Packet are invalid	done
tc_mqtt_broker_connect_002	If the Protocol Name is incorrect the IUT MAY disconnect the Client or it MAY continue processing the CONNECT packet.	done
tc_mqtt_broker_connect_003	The IUT MUST respond to Protocol Levels which it supports (in scope: MQTT-3.1.1) with return code 0x00	done
tc_mqtt_broker_connect_004	The IUT MUST validate that the reserved flag in the CONNECT Control Packet is set to zero and disconnect the Client if it is not zero	done
tc_mqtt_broker_connect_005	If the Will Flag is set to 1, the Will QoS and Will Retain fields in the Connect Flags will be used by the IUT, and the Will Topic and Will Message fields MUST be present in the payload.	done
tc_mqtt_broker_connect_006	If the Will Flag is set to 0 the Will QoS and Will retain fields in the Connect Flags MUST be set to zero and the Will Topic and Will Message fields MUST NOT be present in the payload.	done
tc_mqtt_broker_connect_007	If the Will Flag is set to 0, then the Will QoS MUST be set to 0	done
tc_mqtt_broker_connect_008	If the Will Flag is set to 1, the value of Will QoS can be 0, 1 or 2. It MUST NOT be 3	done
tc_mqtt_broker_connect_009	If the Will Flag is set to 1, the value of Will QoS can be 0, 1 or 2.	done
tc_mqtt_broker_connect_010	If the Will Flag is set to 0, then the Will Retain Flag MUST be set to 0	done
tc_mqtt_broker_connect_011	If the Will Flag is set to 0, then the Will Retain Flag MUST be set to 0	done
tc_mqtt_broker_connect_012	If the User Name Flag is set to 0, the Password Flag MUST be set to 0	done
tc_mqtt_broker_connect_013	If the user name flag is set to 0, a user name MUST NOT be present in the payload	done

tc_mqtt_broker_connect_014	If the User Name Flag is set to 1, a user name MUST be present in the payload	done
tc_mqtt_broker_connect_015	If the Password Flag is set to 0, a password MUST NOT be present in the payload	done
tc_mqtt_broker_connect_016	If the Password Flag is set to 1, a user password MUST be present in the payload	done
tc_mqtt_broker_connect_017	The IUT MUST allow ClientIDs which are between 1 and 23 UTF-8 encoded bytes in length	done
tc_mqtt_broker_connect_018	The IUT MUST allow ClientIDs that contain only alphanumeric characters [0-9a-zA-Z].	done
tc_mqtt_broker_connect_019	The IUT MAY allow a Client to supply a Client Identifier that has a length of zero bytes, however if it does so the IUT MUST treat this as a special case and assign a unique Client Identifier to that Client.	done
tc_mqtt_broker_connect_020	The ClientId MUST be a well-formed UTF-8 encoded String.	none
tc_mqtt_broker_connect_021	The ClientId MUST be a UTF-8 encoded String.	none
tc_mqtt_broker_connect_022	The Will Topic MUST be a UTF-8 encoded string	none
tc_mqtt_broker_connect_023	The Will Topic MUST be a UTF-8 encoded string	none
tc_mqtt_broker_connect_024	The User Name MUST be a UTF-8 encoded string.	none
tc_mqtt_broker_connect_025	The User Name MUST be a UTF-8 encoded string.	none
tc_mqtt_broker_connack_001	The IUT accepts only valid fixed header flags for CONNECT Control Packet and responds with CONNACK	done
tc_mqtt_broker_connack_002	If the IUT accepts a connection with CleanSession set to 1, the IUT MUST set SessionPresent to 0 in the CONNACK packet in addition to setting a zero-return code.	done
tc_mqtt_broker_connack_003	If the IUT has stored session state, it MUST set session present flag to 1 in the CONNACK control packet.	done
tc_mqtt_broker_connack_004	If the IUT does not have a stored session state, it MUST set session present flag to 0 in the CONNACK control packet.	done
tc_mqtt_broker_connack_005	The IUT MUST respond to Protocol Levels which it does not support (in scope: MQTT-3.1.1) with return code 0x01	done
tc_mqtt_broker_connack_006	If the Client supplies a zero-byte Client Identifier with Clean Session set to 0, the IUT MUST respond to the CONNECT Control Packet with a CONNACK return code 0x02 and then close the Network Connection.	done
tc_mqtt_broker_connack_007	The Network Connection has been made but the MQTT service is unavailable	done
tc_mqtt_broker_connack_008	The data in the username or password is malformed	done

tc_mqtt_broker_subscribe_001	The IUT closes the network connection if fixed header flags in SUBSCRIBE Control Packet are invalid	done
tc_mqtt_broker_subscribe_002	SUBSCRIBE Control Packet MUST contain a non-zero 16-bit Packet Identifier	none
tc_mqtt_broker_subscribe_003	The Topic Filters in a Subscribe packet payload MUST be UTF-8 encoded strings. This data MUST NOT include encodings of code points between U+D800 and U+DFFF	none
tc_mqtt_broker_subscribe_004	The Topic Filters in a Subscribe packet payload MUST be UTF-8 encoded strings. A UTF-8 encoded string MUST NOT include an encoding of the null character U+0000.	none
tc_mqtt_broker_subscribe_005	The payload of a SUBSCRIBE packet MUST contain at least one Topic Filter /QoS pair.	done
tc_mqtt_broker_subscribe_006	The upper 6 bits of the Requested QoS byte are reserved. Reserved bits for Requested QoS in the payload must be set to 0.	done
tc_mqtt_broker_subscribe_007	The Requested QoS in the payload MUST be 0, 1 or 2	done
tc_mqtt_broker_suback_001	The IUT accepts only valid fixed header flags for SUBSCRIBE Control Packet and responds with SUBACK	done
tc_mqtt_broker_suback_002	SUBSCRIBE (in case where QoS > 0) Control Packet MUST contain a non-zero 16-bit Packet Identifier	done
tc_mqtt_broker_suback_003	SUBACK return codes other than 0x00, 0x01, 0x02 and 0x80 are reserved and MUST NOT be used.	done
tc_mqtt_broker_suback_004	SUBACK return codes other than 0x00, 0x01, 0x02 and 0x80 are reserved and MUST NOT be used.	done
tc_mqtt_broker_suback_005	SUBACK return codes other than 0x00, 0x01, 0x02 and 0x80 are reserved and MUST NOT be used.	done
tc_mqtt_broker_suback_006	SUBACK return codes other than 0x00, 0x01, 0x02 and 0x80 are reserved and MUST NOT be used.	none
tc_mqtt_broker_047	The IUT closes the network connection if fixed header flags in UNSUBSCRIBE Control Packet are invalid	none
tc_mqtt_broker_048	UNSUBSCRIBE Control Packet MUST contain a non-zero 16-bit Packet Identifier	none
tc_mqtt_broker_049	The Topic Filters in an Unsubscribe packet payload MUST be UTF-8 encoded strings. This data MUST NOT include encodings of code points between U+D800 and U+DFFF	none
tc_mqtt_broker_050	The payload of a UNSUBSCRIBE packet MUST contain at least one Topic Filter.	none
tc_mqtt_broker_051	The IUT accepts only valid fixed header flags for UNSUBSCRIBE Control Packet and responds with SUBACK	none

tc_mqtt_broker_052	The IUT MUST respond to an UNSUBSCRIBE request by sending an UNSUBACK control packet. The UNSUBACK control packet MUST have the same Packet Identifier as the UNSUBSCRIBE Control Packet.	none
tc_mqtt_broker_053	The IUT closes the network connection if fixed header flags in PINGREQ Control Packet are invalid	none
tc_mqtt_broker_054	The IUT accepts only valid fixed header flags for PINGREQ Control Packet and responds with a PINGRESP	none
tc_mqtt_broker_055	The IUT MUST validate that reserved bits are set to zero and disconnect the Client if they are not zero.	none
tc_mqtt_broker_056	The IUT MUST validate that reserved bits are set to zero and disconnect the Client if they are not zero.	none
tc_mqtt_broker_057	Retained messages do not form part of the Session state in the IUT, they MUST NOT be deleted when the Session ends	none
tc_mqtt_broker_058	If the Retain Flag is set to 1, in a PUBLISH Control Packet sent by a Client to the IUT, the IUT MUST store the Application Message and its QoS, so that It can be delivered to future subscribers.	none
tc_mqtt_broker_059	The IUT MUST set the Retain Flag to 0 when a PUBLISH Control Packet is sent to a Client because it matches an established subscription regardless of how the flag was set in the message it received.	none
tc_mqtt_broker_060	A PUBLISH Control Packet with a Retain Flag set to 1 and a payload containing zero bytes will be processed as normal by the IUT and sent to Clients with a subscription matching the topic name. Additionally, any existing retained message with the same topic name MUST be removed and any future subscribers for the topic will not receive a retained message.	none
tc_mqtt_broker_061	If the Retain Flag is 0, in a PUBLISH Control Packet sent by a Client to the IUT, the IUT MUST NOT store the message and MUST NOT remove or replace any existing retained message.	none
tc_mqtt_broker_062	If the IUT receives a QoS 0 message with a Retain Flag set to 1 it MUST discard any message previously retained for that topic.	none
tc_mqtt_broker_063	If the User Name and Password Flags are set to 1, then User Name and Password MUST be present in the payload.	none
tc_mqtt_broker_064	The IUT expects only a CONNECT packet as the first packet for each new connection	none
tc_mqtt_broker_065	The IUT expects only a CONNECT packet as the first packet for each new connection. Interaction with the IUT before a CONNECT is not allowed.	none
tc_mqtt_broker_066	The IUT can only receive exactly one CONNECT packet over a network connection.	none
tc_mqtt_broker_067	The IUT must be able to restore a session for a client	none

tc_mqtt_broker_068	If CleanSession is set to 1 the IUT MUST discard any previous sessions for this client and start a new one.	none
tc_mqtt_broker_069	If the IUT receives a SUBSCRIBE packet containing a Topic Filter that is identical to an existing Subscription's Topic Filter then it MUST completely replace that existing Subscription.	none
tc_mqtt_broker_070	If the IUT receives a SUBSCRIBE packet containing a Topic Filter that is identical to an existing Subscription's Topic Filter then it MUST completely replace that existing Subscription. Any existing retained messages matching the Topic Filter MUST be re-sent.	none
tc_mqtt_broker_071	If the IUT receives a SUBSCRIBE packet containing multiple Topic Filters it MUST handle that packet as if it had received a sequence of multiple SUBSCRIBE packets, except that it combines their responses into a single SUBACK response.	none
tc_mqtt_broker_072	UNSUBSCRIBE (in case where QoS > 0) Control Packet MUST contain a non-zero 16-bit Packet Identifier	none
tc_mqtt_broker_073	UNSUBSCRIBE (in case where QoS > 0) Control Packet MUST contain a non-zero 16-bit Packet Identifier	none
tc_mqtt_broker_074	If an IUT receives an UNSUBSCRIBE packet that contains multiple Topic Filters it MUST handle that packet as if it had received a sequence of multiple UNSUBSCRIBE packets, except that it sends just one UNSUBACK response.	none
tc_mqtt_broker_075	Clients can send further Control Packets immediately after sending a CONNECT Control Packet. If the IUT rejects the CONNECT, it MUST NOT process any data sent by the Client after the CONNECT Control Packet	none
tc_mqtt_broker_076	A Will Message MUST be stored on the IUT and associated with the Network Connection. The Will Message MUST be published when the network connection is closed before a DISCONNECT Packet.	none
tc_mqtt_broker_077	When the IUT receives a DISCONNECT Control Packet it MUST discard any Will Message associated with the current connection without publishing it.	none
tc_mqtt_broker_078	The Will Message MUST be removed from the stored Session state in the IUT once it has been published	none
tc_mqtt_broker_079	If the Will Flag is set to 1 and Will Retain is set to 1, the IUT MUST publish the Will Message as a retained message.	none
tc_mqtt_broker_080	If the Will Flag is set 0, a Will Message MUST NOT be published when this Network Connection ends.	none
tc_mqtt_broker_081	If the Keep Alive value is non-zero and the IUT does not receive a Control Packet from the Client within $\$T_KA * 1.5$ it MUST disconnect the Network Connection to the Client as if the network had failed.	none
tc_mqtt_broker_082	If the Keep Alive value is non-zero and the IUT does not receive a Control Packet from the Client within $\$T_KA * 1.5$	none

	it MUST disconnect the Network Connection to the Client as if the network had failed.	
tc_mqtt_broker_083	The first packet sent from the Server to the IUT MUST be a CONNACK packet.	none
tc_mqtt_broker_084	The first packet sent from the Server to the IUT MUST be a CONNACK packet.	none
tc_mqtt_broker_085	The first packet sent from the Server to the IUT MUST be a CONNACK packet.	none
tc_mqtt_broker_086	The DUP flag MUST be set to 0 for all QoS 0 messages	none
tc_mqtt_broker_087	The DUP flag MUST be set to 0 for all QoS 0 messages	none
tc_mqtt_broker_088	The DUP flag MUST be set to 1 when a re-delivery of a PUBLISH Control Packet is attempted.	none
tc_mqtt_broker_089	A PUBLISH Control Packet MUST NOT have QoS Level set to 3.	none
tc_mqtt_broker_090	The Topic Name MUST be present as the first field in the PUBLISH Control Packet variable header. It MUST be a UTF-8 encoded string.	none
tc_mqtt_broker_091	The Topic Name in the PUBLISH Control Packet MUST NOT contain wildcard characters.	none
tc_mqtt_broker_092	If the IUT receives a PUBLISH Control Packet with QoS 1, it MUST respond with a PUBACK Control Packet.	none
tc_mqtt_broker_093	If the IUT receives a PUBLISH Control Packet with QoS 2, it MUST respond with a PUBREC Control Packet.	none
tc_mqtt_broker_094	The IUT MUST respond with PUBACK Control Packet containing the Packet Identifier from the incoming PUBLISH Control Packet, having accepted ownership of the Application Message. After the IUT has sent a PUBACK it MUST treat any incoming PUBLISH Control Packets that contain the same Packet Identifier as being a new publication, irrespective of the setting of its DUP flag.	none
tc_mqtt_broker_095	The IUT MUST assign an unused Packet Identifier when it has a new Application Message to publish and send a PUBLISH Control Packet with QoS=2 and DUP=0	none
tc_mqtt_broker_096	The value of the DUP flag from an incoming PUBLISH Control Packet is not propagated when the PUBLISH Control Packet is sent to subscribers by the IUT. The DUP flag in the outgoing PUBLISH Control Packet is set independently to the incoming PUBLISH Control Packet, its value MUST be determined solely by whether the outgoing PUBLISH Control Packet is a retransmission.	none
tc_mqtt_broker_097		none
tc_mqtt_broker_098	It is the responsibility of the Client to ensure that the interval between Control Packets being sent does not exceed the Keep Alive value. In the absence of sending any other Control Packets, the Client MUST send a PINGREQ Packet	none

Date : Tue Jul 11 15:32:05 2017