

VERSION 1.6
MAY 26, 2020

MQTT CONNECTOR PROFESSIONAL

SW MANUAL
Developer guide



4EACH S.R.O.
WWW.4EACH.CZ

MQTT CONNECTOR PROFESSIONAL API REFERENCE

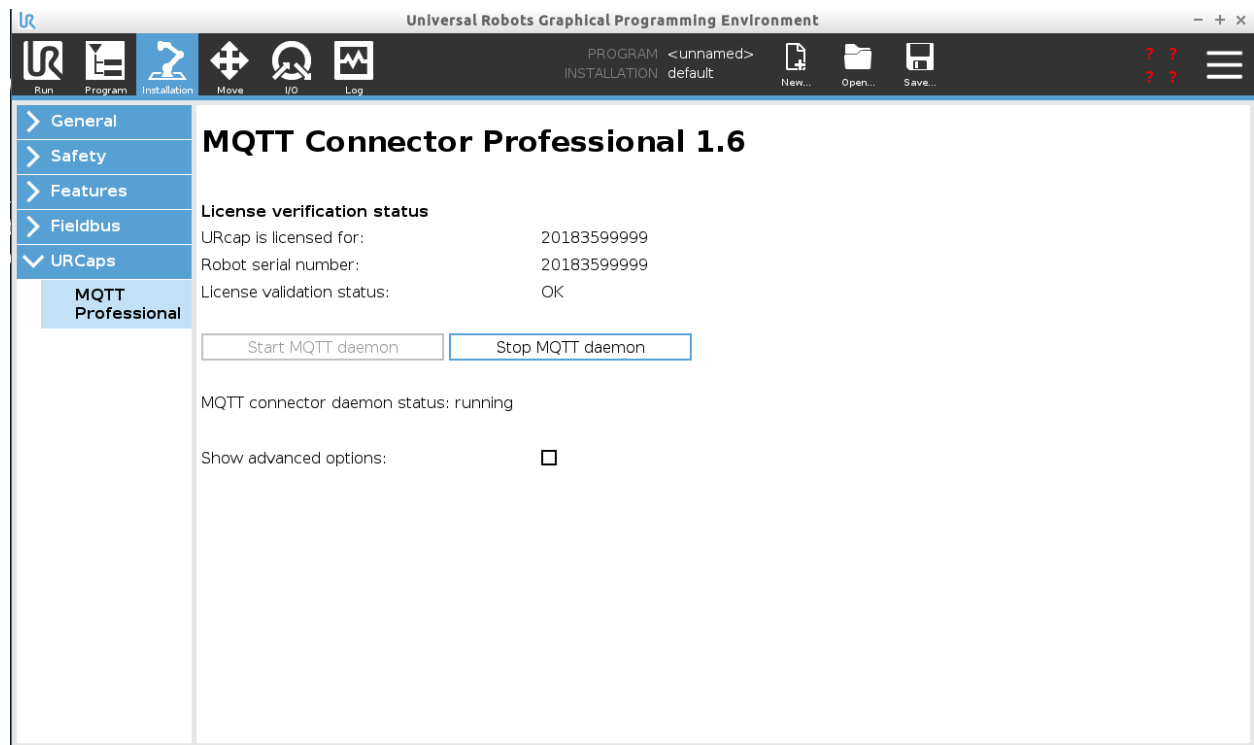
Documentation valid for URCap version 1.6.0

RELEASE NOTES

This version completely changed the underlying technology. This change significantly improved the stability of the product especially on eSeries.

Although we spent a lot of effort to keep the product fully compatible with previous versions, some error codes were added and the function `publish_on_stop` doesn't have the timeout parameter anymore.

INSTALLATION TAB



Installation tab incorporates buttons to start and stop MQTT daemon. Also licensing details are displayed.

Physical robots require purpose built URCap with correct serial number.

Under advanced options, error logging priority selections is present. In default state standard option is selected.

- Standard log priority – logs only errors and critical information
- Debug log priority – logs all possible information useful for debugging and diagnostics

Note, that log priority is set on daemon start. To change priority from standard to debug, changing dropdown selection and restarting daemon is required. This state is applied until next restart, which will use default standard log priority to ensure optimal performance. After robot power cycle, standard log priority is applied.

MQTT_INITIALIZE

initializes connection details using credentials.

It takes the following arguments:

- Hostname – the hostname or IP address of the remote broker [string]
- Port - the network port of the server host to connect to [int]
- User – username to access to remote broker [string]
- Password - password to access to remote broker [string]

Note, that some brokers will accept anonymous users, even though they provide username and password.

MQTT_INITIALIZE_ANONYMOUS

initializes connection details using anonymous access (without username and password).

It takes the following arguments:

- Hostname – the hostname or IP address of the remote broker [string]
- Port - the network port of the server host to connect to [int]

MQTT_SET_MAX_QUEUE

Sets maximum outgoing messages queue length. This is important in case of connection loss, because every queued message is send after connection is restored. This can cause temporary network overload. Parameter with value 0 sets unlimited queue length. When the queue is full, any further outgoing messages would be dropped.

- Length - maximum outgoing messages queue length [int]

MQTT_SET_LAST_WILL

Set a Will to be sent to the broker. If the client (ungracefully) disconnects without calling disconnect(), the broker will publish the message on its behalf.

- Topic - the topic that the will message should be published on [string]
- Message - the message to send as a will [string]
- Qos - the quality of service level to use for the will [int]
- Retained - if set to True, the will message will be set as the “last known good”/retained message for the topic [bool]

Note, that this function must be called after initialize function and before connect function.

MQTT_SET_PUBLISH_ON_STOP

Set a message to be sent to the broker when UR program stops.

- Topic - the topic that the will message should be published on [string]
- Message - the message to send as a will [string]
- Qos - the quality of service level to use for the will [int]
- Retained - if set to True, the will message will be set as the “last known good”/retained message for the topic [bool]

Timeout parameter used in previous versions of the product is deprecated. Message is always send until 10 seconds after the program is stopped.

Note, that this function differs from last will. This function sends specified messages when the robot program changes the state from running to stopped. This can occur by user interaction; emergency stop or other factors. Messages to be send are stored locally, therefore if ungraceful disconnect occurs, no messages can be sent. Message buffer is limited to 100 entries.

MQTT_CONNECT

Connects the client to a broker, using default timeout period of 5 seconds.

MQTT_CONNECT_TIMEOUT

Connects the client to a broker, using custom timeout period defined by parameter.

- Timeout - Ensures maximum blocking duration of function call. Time is defined in milliseconds, and changes in 100ms increments. [int]

MQTT_SUBSCRIBE

Subscribe the client to one topic.

- Topic - subscription topic to subscribe to [string]

MQTT_UNSUBSCRIBE

Unsubscribe the client from one topic.

- Topic - subscription topic to unsubscribe from [string]

MQTT_UNSUBSCRIBE_ALL

Unsubscribe the client from all topics.

MQTT_GET_MESSAGE

Function used to lookup incoming messages buffer. Returns latest received value from given topic.

- Topic – Specifies topic to search for [string]

Note, that this function has different communication interface than other methods. It doesn't return OK status code, but actual message payload. If error is present, error codes are returned.

MQTT_PUBLISH

This function causes a message to be sent to the broker and subsequently from the broker to any clients subscribing to matching topics, using default timeout period of 3 seconds. It takes the following arguments:

- Topic - the topic that the will message should be published on [string]
- Message - the message to send as a will [string]
- Qos - the quality of service level to use for the will [int]
- Retained - if set to True, the will message will be set as the “last known good”/retained message for the topic [bool]

MQTT_PUBLISH_TIMEOUT

This function causes a message to be sent to the broker and subsequently from the broker to any clients subscribing to matching topics, using custom timeout period defined by parameter. It takes the following arguments:

- Topic - the topic that the will message should be published on [string]
- Message - the message to send as a will [string]
- Qos - the quality of service level to use for the will [int]
- Retained - if set to True, the will message will be set as the “last known good”/retained message for the topic [bool]
- Timeout - Ensures maximum blocking duration of function call. Time is defined in milliseconds, and changes in 100ms increments. [int]

MQTT_DISCONNECT

Disconnects from the broker cleanly. Using disconnect() will not result in a will message being sent by the broker. Disconnect will not wait for all queued message to be sent.

DEBUGGING PRACTICE AND ERROR CODES

To ease programing, error troubleshooting and general ease of use, MQTT Connector uses error codes as return values for most of its functions. This can be helpful in situations where client-broker connectivity is not stable, or during initial setup.

General recommended usage

- Compare return values against constants or predefined variables using IF statement. Return values should be compared against error codes table.
- Store return values in variables. Even though value is never compared against error codes, it can be helpful to view variable values in Polyscope graphical interface.

ERROR CODE TABLE

XMLRPC Function	Description	Status	Error code
ALL FUNCTIONS	NOT AUTHORIZED	NG	NG-CODE-001
authorize	SUCCESS	OK	OK-CODE-002
	UNKNOWN ERROR	NG	NG-CODE-003
ALL FUNCTIONS	<i>deprecated</i>	NG	NG-CODE-004
	DAEMON NOT READY – INTERNAL ERROR	NG	NG-CODE-005
	CLIENT NOT INITIALIZED – call initialize	NG	NG-CODE-006
initialize	SUCCESS	OK	OK-CODE-010
	FAILED	NG	NG-CODE-011
connect	MQTT CONNECT OK	OK	OK-CODE-020
	NO ROUTE TO HOST	NG	NG-CODE-021
	FAILED - timeout	NG	NG-CODE-022
	UNKNOWN ERROR	NG	NG-CODE-023
disconnect	MQTT DISCONNECT OK	OK	OK-CODE-030
	UNKNOWN ERROR	NG	NG-CODE-031
subscribe	SUBSCRIBE OK	OK	OK-CODE-040
	SUBSCRIBE FAILED - already exists	NG	NG-CODE-041
	UNKNOWN ERROR	NG	NG-CODE-042
unsubscribe	UNSUBSCRIBE OK	OK	OK-CODE-050
	UNSUBSCRIBE FAILED - no such subscription	NG	NG-CODE-051
	UNKNOWN ERROR	NG	NG-CODE-052
unsubscribe_all	UNSUBSCRIBE ALL OK	OK	OK-CODE-060
	UNKNOWN ERROR	NG	NG-CODE-061
get_message	MESSAGE PAYLOAD STRING	OK	
	NOT YET RECEIVED ANY MESSAGE	NG	NG-CODE-071
	NO SUCH SUBSCRIPTION	NG	NG-CODE-072
	FALSE PARAMS	NG	NG-CODE-073
	UNKNOWN ERROR	NG	NG-CODE-074
publish	SUCCESS	OK	OK-CODE-080
	FAILED	NG	NG-CODE-081
	MESSAGE SEND NOK - broker disconnected	NG	NG-CODE-082
	FALSE PARAMS	NG	NG-CODE-083
	UNKNOWN ERROR	NG	NG-CODE-084
set_max_queued_messages	SUCCESS	OK	OK-CODE-090
	FAILED	NG	NG-CODE-091
set_last_will	SUCCESS	OK	OK-CODE-100
	FAILED	NG	NG-CODE-101
set_publish_on_stop	SUCCESS	OK	OK-CODE-110
	FALSE PARAMS	NG	NG-CODE-111
	BUFFER IS FULL	NG	NG-CODE-112
ALL FUNCTIONS	UNKNOWN ERROR	NG	NG-CODE-113
ALL FUNCTIONS	FULL TASK QUEUE		NG-CODE-991
ALL FUNCTIONS	NOT EXISTING TASK ID		NG-CODE-992
ALL FUNCTIONS	TASK NOT FINISHED		NG-CODE-993
ALL FUNCTIONS	UNKNOWN ENQUEUE EXCEPTION		NG-CODE-994
ALL FUNCTIONS	UNKNOWN RESULT EXCEPTION		NG-CODE-995
ALL FUNCTIONS	FUNCTION CALL TIMEOUT		NG-CODE-996