
mqtt-codec Documentation

Release 1.0.2

Keegan Callin

Jan 31, 2019

Contents:

1	Installation	3
2	Project Infrastructure	5
3	Table of Contents	7
3.1	User Guide	7
3.2	API Reference	9
3.3	Change Log	28
3.4	Developer Guide	29
3.5	Distribution Procedure	32
4	Indices and tables	37
	Python Module Index	39

The *mqtt-codec* package is an MQTT packet encoder and decoder (codec). The library has high test coverage (~94%) and is known to perform well in distributed IoT networks with thousands of nodes.

CHAPTER 1

Installation

The mqtt-codec package is distributed through pypi.org and can be installed with the standard Python package manager [pip](#):

```
$ pip install mqtt-codec
```

If you do not have pip then the package can be downloaded from [mqtt-codec](#) and installed with the standard *setup.py* method:

```
$ python setup.py install
```


CHAPTER 2

Project Infrastructure

The project is coordinated through public infrastructure:

- Releases ([pypi](#))
- Documentation ([readthedocs.io](#))
- Bug Tracker ([github](#))
- Code Repository ([github](#))

3.1 User Guide

The *mqtt_codec* package is a “weapons grade” stateless package for encoding and decoding MQTT 3.1.1 packets as defined in the specification at <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf>.

The package is hyperbolically “weapons grade” because its primary goal is thoroughness of implementation and reliability. Speed is a distant secondary priority; having said this, nobody has ever lodged a complaint about the library being too slow.

3.1.1 Usage

A basic encode/decode cycle looks like this:

```
>>> from io import BytesIO
>>> from binascii import b2a_hex
>>> import mqtt_codec.packet
>>> import mqtt_codec.io
>>>
>>> # Encode a Connect packet
>>> will = mqtt_codec.packet.MqttWill(qos=0, topic='hello', message='message',
↳retain=True)
>>> connect = mqtt_codec.packet.MqttConnect(client_id='client_id', clean_
↳session=False, keep_alive=0, will=will)
>>> with BytesIO() as f:
...     num_bytes_written = connect.encode(f)
...     buf = f.getvalue()
...
>>> assert len(buf) == num_bytes_written
>>> print('0x{} ({} bytes)'.format(b2a_hex(buf), len(buf)))
0x102500044d515454042400000009636c69655e745f6964000568656c6c6f00076d657373616765 (39
↳bytes)
>>>
```

(continues on next page)

(continued from previous page)

```
>>> # Decode the connect packet and assert equality.
>>> with mqtt_codec.io.BytesReader(buf) as f:
...     num_bytes_read, decoded_connect = connect.decode(f)
...
>>> assert len(buf) == num_bytes_written
>>> assert connect == decoded_connect
>>> print(' Encoded {}'.format(connect))
    Encoded MqttConnect(client_id='client_id', clean_session=False, keep_alive=0,
↳username=***, password=***, will=MqttWill(topic=hello, payload=0x6d657373616765,
↳retain=True, qos=0))
>>> print('= Decoded {}'.format(decoded_connect))
= Decoded MqttConnect(client_id=u'client_id', clean_session=False, keep_alive=0,
↳username=***, password=***, will=MqttWill(topic=hello, payload=0x6d657373616765,
↳retain=True, qos=0))
```

3.1.2 Requirements

The `mqtt-codec` project has been tested on Linux against these environments:

- Python 2.7
- Python 3.4
- Python 3.5
- Python 3.6
- Python 3.7

The codec likely works on Python 3.0 - 3.3 as well but these tests are not part of the standard docker container test suite.

Package Dependencies

When running Python versions less than 3.4 the `enum34` package is required. Besides there are no other required packages.

Processor and Memory Usage

The maximum size of an MQTT packet is `mqtt_codec.packet.MqttFixedHeader.MAX_REMAINING_LEN` (=268435455 bytes). Encoding or decoding an mqtt message may consume up to this many bytes. Smaller messages require less memory to encode or decode.

While constructing an MQTT packet it is necessary to temporarily encode it so that the byte size of contained UTF-8 strings can be determined and the final packet size calculated. This means that constructing an MQTT packet can temporarily consume up to `mqtt_codec.packet.MqttFixedHeader.MAX_REMAINING_LEN` (=268435455 bytes) of memory and a proportionate amount of processor time. In practice most packets tend to be much smaller than this and the processor time seems small enough for most applications.

3.1.3 Testing and Quality

The `mqtt-codec` package is tested against most use and abuse cases. It has proven itself in distributed IoT environments with thousands of nodes and expected to perform as well or better than most quality industrial scale systems. There is

a high bar to marking a release as stable and it usually takes more than a month of field data collection on a prospective release before this happens.

The codec has not proven itself in hostile and malicious environments and has not seen thorough 3rd-party review from a security specialist. If you are interested in assisting then please contact the author, [Keegan Callin](#).

3.1.4 Semantic Versioning

The *mqtt-codec* package is versioned according to [Semantic Versioning 2.0.0](#) guidelines. A summary of SemVer is included here for your convenience:

Given a version number MAJOR.MINOR.PATCH, increment the:

1. MAJOR version when you make incompatible API changes,
2. MINOR version when you add functionality in a backwards-compatible manner, and
3. PATCH version when you make backwards-compatible bug fixes.

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.

—Semantic Versioning Summary, <<https://semver.org/#summary>>, retrieved 2018-10-01.

3.1.5 Bugs and Enhancements

As the maintainer of this library I, [Keegan Callin](#), welcome your polite, constructive comments and criticisms of this library at the [github issue tracker](#).

3.2 API Reference

3.2.1 *mqtt_codec.packet* Package

A collection of classes used to represent MQTT control packets as described in the specification. The classes are, in general, immutable; once a class has been instantiated its properties cannot be changed.

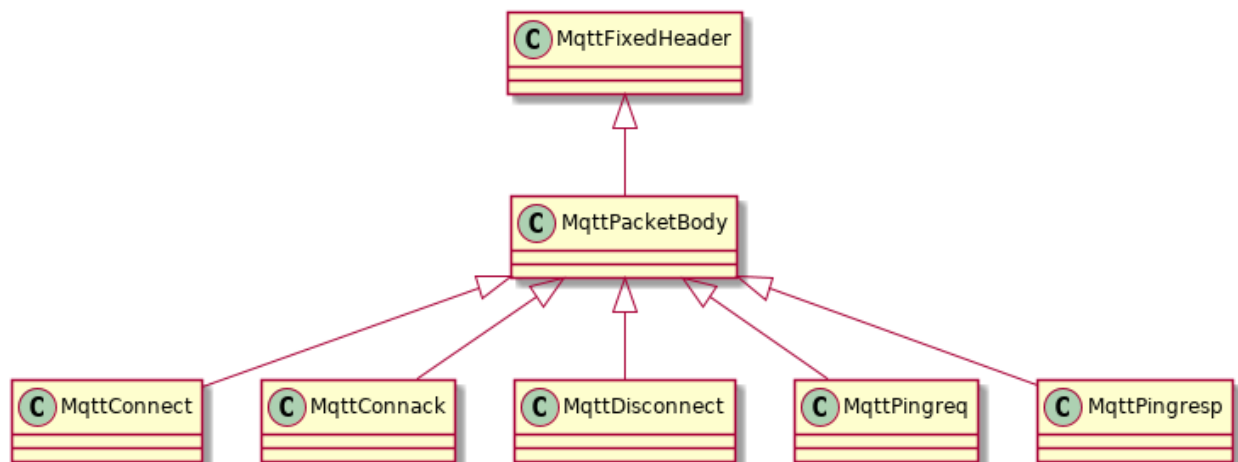


Fig. 1: Connection Packets

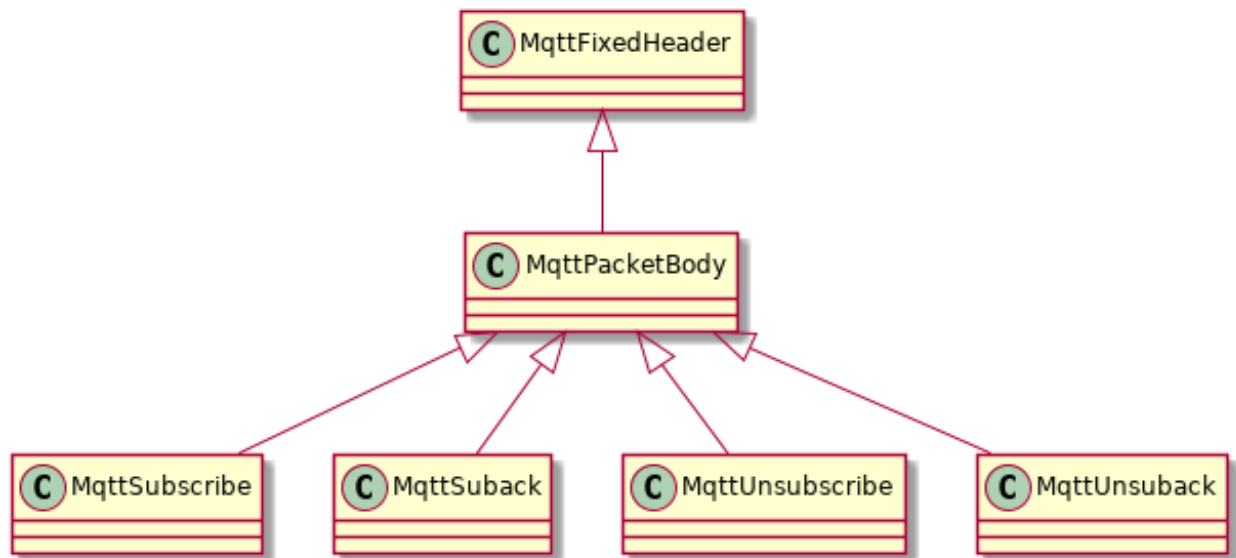


Fig. 2: Subscrube/Unsubscribe Packets

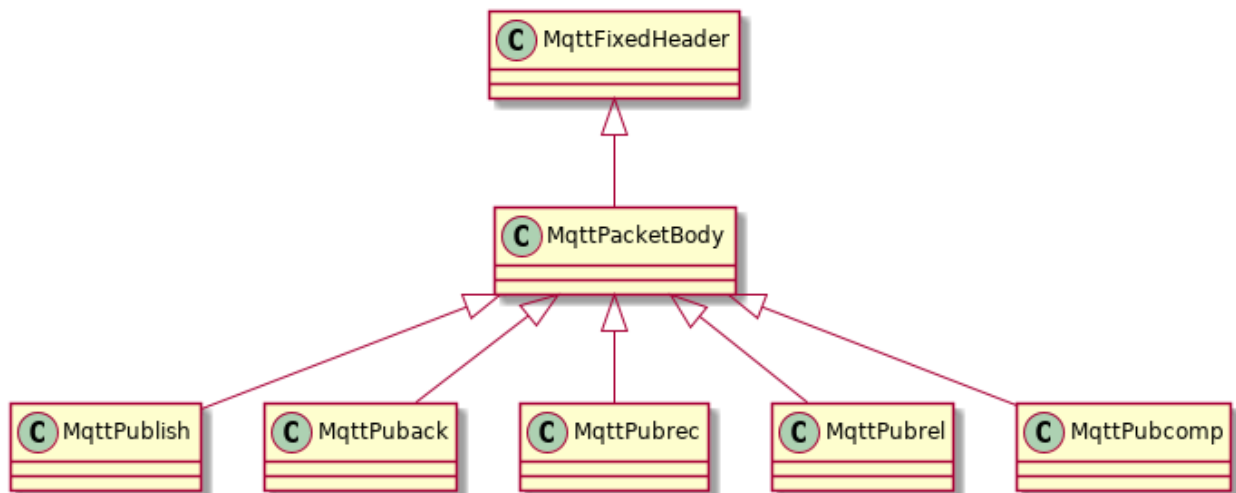


Fig. 3: Publish Packets

class mqtt_codec.packet.ConnackResult

Bases: enum.IntEnum

ConnackResult codes as enumerated in Table 3.1 (line 709) of the MQTT 3.1.1 specification.

accepted = 0

fail_bad_client_id = 2

fail_bad_protocol_version = 1

fail_bad_username_or_password = 4

fail_not_authorized = 5

fail_server_unavailable = 3

class mqtt_codec.packet.MqttConnack(*session_present*, *return_code*)

Bases: mqtt_codec.packet.MqttPacketBody

An immutable representation of an MQTT Connack packet as described in MQTT 3.2 (line 655).

Parameters

- **session_present** (*bool*) – Session present.
- **return_code** (*ConnackResult*) –

classmethod decode_body(*header*, *f*)

Parameters

- **header** (*MqttFixedHeader*) –
- **f** (*file*) – Object with a read method.

Raises

- *DecodeError* – When bytes have values incompatible with a MqttConnack packet.
- *UnderflowDecodeError* – When not enough bytes are available to decode a complete packet.

Returns

- *int* – Number of bytes consumed from *f*.
- *MqttConnack* – Object extracted from *f*.

encode_body (*f*)

Parameters *f* (*file*) – File-like object with write method.

Returns Number of bytes written to file.

Return type *int*

return_code

ConnackResult – Result of the connect as described in MQTT 3.2.2.3 line 701.

session_present

bool – Session present flag as described in MQTT 3.2.2.2 line 676.

class mqtt_codec.packet.MqttConnect(*client_id*, *clean_session*, *keep_alive*, *username=None*,
password=None, *will=None*)

Bases: mqtt_codec.packet.MqttPacketBody

An immutable representation of an MQTT connect object as in MQTT 3.1 (line 364).

The value of `str(self)` will have the username and password obscured so that it can be placed in logfiles without compromising the connection username and password. The value or `repr(self)` does not obscure the username and password.

Raises `mqtt_codec.io.TooBigEncodeError` – The parameters are impossible large to create an MQTT packet for. It encoded length must be greater than `MqttFixedHeader.MAX_REMAINING_LEN` (=268435455 bytes) in order to cause this error.

Parameters

- **client_id** (*str*) –
- **clean_session** (*bool*) –
- **keep_alive** (*int*) – $0 \leq \text{keep_alive} \leq 2^{16}-1$
- **username** (*str or None*) –
- **password** (*str or None*) –
- **will** (*MqttWill or None*) –

CONNECT_HEADER = `'\x00\x04MQTT'`

PROTOCOL_LEVEL = `'\x04'`

clean_session
bool – MQTT password.

client_id
str – Client id.

classmethod decode_body (*header, f*)

Parameters

- **header** (*MqttFixedHeader*) –
- **f** (*file*) – File-like object with a read method.

Returns

- *int* – Number of bytes consumed from *f*.
- *MqttConnect* – Object extracted from *f*.

encode_body (*f*)

Parameters *f* (*file*) – File-like object with a write method.

Returns Number of bytes written to *f*.

Return type *int*

keep_alive
int – Keep alive period as described in MQTT 3.1.1 specification 3.1.2.10. When zero keep-alive is disabled. If positive then after *self.keep_alive* seconds of inactivity the client will send a ping to the server.

password
str or None – MQTT password.

username
str or None – MQTT username.

will
MqttWill or None – A message that will be published on behalf of the client by the server in case of an unexpected disconnect. If *None* then the server does not publish any message on behalf of the client.


```
class mqtt_codec.packet.MqttControlPacketType
```

Bases: `enum.IntEnum`

An enumeration of MQTT control packet types as described in the MQTT 3.1.1 specification in Table 2.1 (line 239).

`connack = 2`

`connect = 1`

`disconnect = 14`

`pingreq = 12`

`pingresp = 13`

`puback = 4`

`pubcomp = 7`

`publish = 3`

`pubrec = 5`

`pubrel = 6`

`suback = 9`

`subscribe = 8`

`unsuback = 11`

`unsubscribe = 10`

```
class mqtt_codec.packet.MqttDisconnect
```

Bases: `mqtt_codec.packet.MqttPacketBody`

An immutable representation of an MQTT Disconnect packet as described in MQTT 3.14 (line 1138).

```
classmethod decode_body(header, f)
```

Generates a `MqttDisconnect` packet given a `MqttFixedHeader`. This method asserts that `header.packet_type` is `MqttControlPacketType.disconnect`.

Parameters

- **header** (`MqttFixedHeader`) –
- **f** (`file`) – Object with a read method.

Raises `DecodeError` – When there are extra bytes at the end of the packet.

Returns

- `int` – Number of bytes consumed from `f`.
- `MqttDisconnect` – Object extracted from `f`.

```
encode_body(f)
```

Parameters **f** (`file`) – File-like object with write method.

Returns Number of bytes written to file.

Return type `int`

```
class mqtt_codec.packet.MqttFixedHeader(packet_type, flags, remaining_len)
```

Bases: `object`

An immutable class that represents an MQTT fixed header as described in MQTT Version 3.1.1 section 2.2 (line 233).

The serialized byte format is summarized as follows:

	Bit							
	7	6	5	4	3	2	1	0
byte 1	control type				flags			
byte 2	remaining length							

Raises `mqtt_codec.io.TooBigEncodeError` – The *remaining_len* exceeds the maximum of `MqttFixedHeader.MAX_REMAINING_LEN` (=268435455 bytes).

Parameters

- **packet_type** (`MqttControlPacketType`) –
- **flags** (`int`) – An assert statement verifies that `are_flags_valid(packet_type, flags)` is True.
- **remaining_len** (`int`) – Asserted to be $0 \leq \text{remaining_len} \leq \text{MqttFixedHeader.MAX_REMAINING_LEN}$

MAX_REMAINING_LEN = 268435455

static decode (*f*)

Extract a `MqttFixedHeader` from *f*.

Parameters *f* (*file*) – Object with read method.

Raises

- `DecodeError` – When bytes decoded have values incompatible with a `MqttFixedHeader` object.
- `UnderflowDecodeError` – When end-of-stream is encountered before the end of the fixed header.

Returns

- `int` – Number of bytes consumed from *f*.
- `MqttFixedHeader` – Header object extracted from *f*.

encode (*f*)

Parameters *f* (*file*) – file-like object

Returns Number of bytes written.

Return type `int`

flags

int – A value $0 \leq \text{flags} \leq 15$ representing the 4-bit MQTT header flags field. The property is guaranteed to comply with [MQTT-2.2.2-1] requirements based on `self.packet_type`.

packet ()

packet_type

`MqttControlPacketType` – MQTT packet type of `self`.

remaining_len

int – Number bytes in packet that follow the packet header.

size
int – Number bytes required to encode the packet *self*.

class mqtt_codec.packet.**MqttPacketBody** (*packet_type*, *flags*)

Bases: *mqtt_codec.packet.MqttFixedHeader*

Raises mqtt_codec.io.TooBigEncodeError – The message body is impossibly large to create an MQTT packet for. It must be greater than *MqttFixedHeader.MAX_REMAINING_LEN* (=268435455 bytes) in order to cause this error.

Parameters

- **packet_type** (*MqttControlPacketType*) –
- **flags** (*int*) – Flags $0 \leq \text{flags} \leq 2^{*8}-1$.

classmethod **decode** (*f*)

Parameters *f* (*file*) – Object with a read method.

Returns

- *int* – Number of bytes consumed from *f*.
- *MqttFixedHeader* – Header object extracted from *f*.

classmethod **decode_body** (*header*, *buf*)

encode (*f*)

Parameters *f* (*file*) – file-like object

Returns Number of bytes written.

Return type *int*

encode_body (*f*)

class mqtt_codec.packet.**MqttPingreq**

Bases: *mqtt_codec.packet.MqttPacketBody*

An immutable representation of an MQTT Pingreq packet as described in MQTT 3.12 (line 1109).

classmethod **decode_body** (*header*, *f*)

Generates a *MqttPingreq* packet given a *MqttFixedHeader*. This method asserts that *header.packet_type* is *pingreq*.

Parameters

- **header** (*MqttFixedHeader*) –
- *f* (*file*) – Object with a read method.

Raises *DecodeError* – When there are extra bytes at the end of the packet.

Returns

- *int* – Number of bytes consumed from *f*.
- *MqttPingreq* – Object extracted from *f*.

encode_body (*f*)

Parameters *f* (*file*) – File-like object with write method.

Returns Number of bytes written to file.

Return type *int*

class mqtt_codec.packet.MqttPingresp

Bases: *mqtt_codec.packet.MqttPacketBody*

An immutable representation of an MQTT Pingresp packet as described in MQTT 3.13 (line 1126).

classmethod decode_body(*header, f*)

Generates a *MqttPingresp* packet given a *MqttFixedHeader*. This method asserts that header.packet_type is *pingresp*.

Parameters

- **header** (*MqttFixedHeader*) –
- **f** (*file*) – Object with a read method.

Raises *DecodeError* – When there are extra bytes at the end of the packet.

Returns

- *int* – Number of bytes consumed from *f*.
- *MqttPingresp* – Object extracted from *f*.

encode_body (*f*)

Parameters **f** (*file*) – File-like object with write method.

Returns Number of bytes written to file.

Return type *int*

class mqtt_codec.packet.MqttPuback(*packet_id*)

Bases: *mqtt_codec.packet.MqttPacketBody*

An immutable representation of an MQTT Puback packet as described in MQTT 3.4 (line 838).

Parameters **packet_id** (*int*) – $0 \leq \text{packet_id} \leq 2^{16} - 1$

classmethod decode_body(*header, f*)

Generates a *MqttPuback* packet given a *MqttFixedHeader*. This method asserts that header.packet_type is *puback*.

Parameters

- **header** (*MqttFixedHeader*) –
- **f** (*file*) – Object with a read method.

Raises *DecodeError* – When there are extra bytes at the end of the packet.

Returns

- *int* – Number of bytes consumed from *f*.
- *MqttPuback* – Object extracted from *f*.

encode_body (*f*)

Parameters **f** (*file*) – File-like object with write method.

Returns Number of bytes written to file.

Return type *int*

packet_id

int – packet id such that $0 \leq \text{packet_id} \leq 2^{16} - 1$.

class mqtt_codec.packet.**MqttPubcomp** (*packet_id*)

Bases: *mqtt_codec.packet.MqttPacketBody*

An immutable representation of MQTT Pubrec packet as described in MQTT 3.7 (line 890).

Parameters *packet_id* (*int*) – $0 \leq \text{packet_id} \leq 2^{16} - 1$

classmethod **decode_body** (*header*, *f*)

Generates a *MqttPubcomp* packet given a *MqttFixedHeader*. This method asserts that *header.packet_type* is *pubcomp*.

Parameters

- **header** (*MqttFixedHeader*) –
- **f** (*file*) – Object with a read method.

Raises *DecodeError* – When there are extra bytes at the end of the packet.

Returns

- *int* – Number of bytes consumed from *f*.
- *MqttPubcomp* – Object extracted from *f*.

encode_body (*f*)

Parameters **f** (*file*) – File-like object with write method.

Returns Number of bytes written to file.

Return type *int*

packet_id

int – packet id such that $0 \leq \text{packet_id} \leq 2^{16} - 1$.

class mqtt_codec.packet.**MqttPublish** (*packet_id*, *topic*, *payload*, *dupe*, *qos*, *retain*)

Bases: *mqtt_codec.packet.MqttPacketBody*

An immutable representation of an MQTT Publish packet as described in MQTT 3.3 (line 715).

Raises *mqtt_codec.io.TooBigEncodeError* – The encoded length of parameters is too long to create an MQTT packet for. The encoded length must be greater than *MqttFixedHeader.MAX_REMAINING_LEN* (=268435455 bytes) in order to cause this error.

Shorten the payload or topic to allow the message to fit.

Parameters

- **packet_id** (*int*) – Integer such that $0 \leq \text{packet_id} \leq (2^{16}) - 1$.

When this object is constructed by the decoder and the QoS is 0 then the decoder will assign this property a value of zero. This property is not used by the encoder. These behaviours are in accordance with the MQTT 3.1.1 specification at 3.3.1-2.

This property is fully encoded/decoded for QoS=1 and QoS=2, again, according to the MQTT specification.

- **topic** (*str*) –
- **payload** (*bytes*) –
- **dupe** (*bool*) – Represents the DUP flag as described by the MQTT specification:

If the DUP flag is set to 0, it indicates that this is the first occasion that the Client or Server has attempted to send this MQTT PUBLISH Packet. If the DUP flag is set to 1, it indicates that this might be re-delivery of an earlier attempt to send the Packet.

The DUP flag MUST be set to 1 by the Client or Server when it attempts to re-deliver a PUBLISH Packet [MQTT-3.3.1-1]. The DUP flag MUST be set to 0 for all QoS 0 messages [MQTT-3.3.1-2].

The value of the DUP flag from an incoming PUBLISH packet is not propagated when the PUBLISH Packet is sent to subscribers by the Server. The DUP flag in the outgoing PUBLISH packet is set independently to the incoming PUBLISH packet, its value MUST be determined solely by whether the outgoing PUBLISH packet is a retransmission [MQTT-3.3.1-3].

- **qos** (*int*) – $0 \leq \text{qos} \leq 2$
- **retain** (*bool*) –

packet_id

int – Integer such that $0 \leq \text{packet_id} \leq (2^{16})-1$.

When this object is constructed by the decoder and the QoS is 0 then the decoder will assign this property a value of zero. This property is not used by the encoder. These behaviours are in accordance with the MQTT 3.1.1 specification at 3.3.1-2.

This property is fully encoded/decoded for QoS=1 and QoS=2, again, according to the MQTT specification.

topic

str

payload

bytes

dupe

bool – Represents the DUP flag as described by the MQTT specification:

If the DUP flag is set to 0, it indicates that this is the first occasion that the Client or Server has attempted to send this MQTT PUBLISH Packet. If the DUP flag is set to 1, it indicates that this might be re-delivery of an earlier attempt to send the Packet.

The DUP flag MUST be set to 1 by the Client or Server when it attempts to re-deliver a PUBLISH Packet [MQTT-3.3.1-1]. The DUP flag MUST be set to 0 for all QoS 0 messages [MQTT-3.3.1-2].

The value of the DUP flag from an incoming PUBLISH packet is not propagated when the PUBLISH Packet is sent to subscribers by the Server. The DUP flag in the outgoing PUBLISH packet is set independently to the incoming PUBLISH packet, its value MUST be determined solely by whether the outgoing PUBLISH packet is a retransmission [MQTT-3.3.1-3].

qos

int – Integer such that $0 \leq \text{qos} \leq 2$.

retain

bool

classmethod decode_body (*header, f*)

Generates a *MqttPublish* packet given a *MqttFixedHeader*. This method asserts that *header.packet_type* is *publish*.

Parameters

- **header** (*MqttFixedHeader*) –

- `f (file)` – Object with a read method.

Raises `DecodeError` – When there are extra bytes at the end of the packet.

Returns

- `int` – Number of bytes consumed from `f`.
- `MqttPublish` – Object extracted from `f`.

dupe

encode_body (`f`)

Parameters `f (file)` – File-like object with write method.

Returns Number of bytes written to file.

Return type `int`

packet_id

`int` – packet id such that $0 \leq \text{packet_id} \leq 2^{16}-1$.

MQTT does not use this property for QoS=0 packets. The deserializers will set this field to zero for QoS=0 packets. When serializing QoS=0 packets this property's value is not placed in the output.

For non-QoS=0 packets (ie. QoS=1, QoS=2) this property is serialized and deserialized as expected.

payload

qos

retain

topic

class `mqtt_codec.packet.MqttPubrec (packet_id)`

Bases: `mqtt_codec.packet.MqttPacketBody`

An immutable representation of MQTT Pubrec packet as described in MQTT 3.5 (line 853).

Parameters `packet_id (int)` – $0 \leq \text{packet_id} \leq 2^{16}-1$

classmethod `decode_body (header, f)`

Generates a `MqttPubrec` packet given a `MqttFixedHeader`. This method asserts that `header.packet_type` is `pubrec`.

Parameters

- **header** (`MqttFixedHeader`) –
- `f (file)` – Object with a read method.

Raises `DecodeError` – When there are extra bytes at the end of the packet.

Returns

- `int` – Number of bytes consumed from `f`.
- `MqttPubrec` – Object extracted from `f`.

encode_body (`f`)

Parameters `f (file)` – File-like object with write method.

Returns Number of bytes written to file.

Return type `int`

packet_id
int – packet id such that $0 \leq \text{packet_id} \leq 2^{16}-1$.

class mqtt_codec.packet.**MqttPubrel** (*packet_id*)

Bases: *mqtt_codec.packet.MqttPacketBody*

An immutable representation of MQTT Pubrel packet as described in MQTT 3.6 (line 869).

Parameters *packet_id* (*int*) – $0 \leq \text{packet_id} \leq 2^{16}-1$

classmethod **decode_body** (*header*, *f*)

Generates a *MqttPubrel* packet given a *MqttFixedHeader*. This method asserts that header.packet_type is *pubrel*.

Parameters

- **header** (*MqttFixedHeader*) –
- **f** (*file*) – Object with a read method.

Raises *DecodeError* – When there are extra bytes at the end of the packet.

Returns

- *int* – Number of bytes consumed from *f*.
- *MqttPubrel* – Object extracted from *f*.

encode_body (*f*)

Parameters **f** (*file*) – File-like object with write method.

Returns Number of bytes written to file.

Return type *int*

packet_id
int – packet id such that $0 \leq \text{packet_id} \leq 2^{16}-1$.

class mqtt_codec.packet.**MqttSuback** (*packet_id*, *results*)

Bases: *mqtt_codec.packet.MqttPacketBody*

An immutable representation of an MQTT Subscribe packet as described in MQTT 3.9 (line 1007).

Raises *mqtt_codec.io.TooBigEncodeError* – There are too many results to create an MQTT packet for. The encoded length must be greater than *MqttFixedHeader.MAX_REMAINING_LEN* (=268435455 bytes) in order to cause this error.

Parameters

- **packet_id** (*int*) – $0 \leq \text{packet_id} \leq 2^{16}-1$
- **results** (*iterable of SubscribeResult*) –

classmethod **decode_body** (*header*, *f*)

Generates a *MqttSuback* packet given a *MqttFixedHeader*. This method asserts that header.packet_type is *suback*.

Parameters

- **header** (*MqttFixedHeader*) –
- **f** (*file*) – Object with a read method.

Raises *DecodeError* – When there are extra bytes at the end of the packet.

Returns

- *int* – Number of bytes consumed from *f*.
- *MqttSuback* – Object extracted from *f*.

encode_body (*f*)

Parameters *f* (*file*) – File-like object with write method.

Returns Number of bytes written to file.

Return type *int*

packet_id

int – packet_id such that $0 \leq \text{packet_id} \leq 2^{16}-1$.

results

tuple of SubscribeResult – Tuple of return codes specifying the maximum QoS level that was granted in each or fail if the subscription failed.

class mqtt_codec.packet.**MqttSubscribe** (*packet_id*, *topics*)

Bases: *mqtt_codec.packet.MqttPacketBody*

An immutable representation of an MQTT Subscribe packet as described in MQTT 3.8 (line 908).

Raises *mqtt_codec.io.TooBigEncodeError* – The parameters are impossibly large to create an MQTT packet for. The encoded length must be greater than *MqttFixedHeader.MAX_REMAINING_LEN* (=268435455 bytes) in order to cause this error.

Parameters

- **packet_id** (*int*) – $0 \leq \text{packet_id} \leq 2^{16}-1$
- **topics** (*iterable of MqttTopic*) –

classmethod **decode_body** (*header*, *f*)

Generates a *MqttSubscribe* packet given a *MqttFixedHeader*. This method asserts that header.packet_type is *subscribe*.

Parameters

- **header** (*MqttFixedHeader*) –
- *f* (*file*) – Object with a read method.

Raises *DecodeError* – When there are extra bytes at the end of the packet.

Returns

- *int* – Number of bytes consumed from *f*.
- *MqttSubscribe* – Object extracted from *f*.

encode_body (*f*)

Parameters *f* (*file*) – File-like object with write method.

Returns Number of bytes written to file.

Return type *int*

packet_id

int – packet id such that $0 \leq \text{packet_id} \leq 2^{16}-1$.

topics

tuple of MqttTopic – Topics requested in subscribe.

class mqtt_codec.packet.**MqttTopic** (*name*, *max_qos*)

Bases: *object*

Parameters

- **name** (*str*) –
- **max_qos** (*int*) – Maximum qos to be granted by server to client.

max_qos

int – Maximum qos to be granted by server to client.

name

str – Topic name.

class mqtt_codec.packet.**MqttUnsuback** (*packet_id*)

Bases: [mqtt_codec.packet.MqttPacketBody](#)

An immutable representation of an MQTT Unsuback packet as described in MQTT 3.11 (line 1093).

Parameters **packet_id** (*int*) – $0 \leq \text{packet_id} \leq 2^{16}-1$

classmethod **decode_body** (*header, f*)

Generates a *MqttUnsuback* packet given a *MqttFixedHeader*. This method asserts that header.packet_type is *unsuback*.

Parameters

- **header** ([MqttFixedHeader](#)) –
- **f** (*file*) – Object with a read method.

Raises *DecodeError* – When there are extra bytes at the end of the packet.

Returns

- *int* – Number of bytes consumed from *f*.
- *MqttUnsuback* – Object extracted from *f*.

encode_body (*f*)

Parameters **f** (*file*) – File-like object with write method.

Returns Number of bytes written to file.

Return type *int*

packet_id

int – packet id such that $0 \leq \text{packet_id} \leq 2^{16}-1$.

class mqtt_codec.packet.**MqttUnsubscribe** (*packet_id, topics*)

Bases: [mqtt_codec.packet.MqttPacketBody](#)

An immutable representation of MQTT Unsubscribe packet as described in MQTT 3.10 (line 1044).

Raises [mqtt_codec.io.TooBigEncodeError](#) – The encoded length of topic parameters is too long to create an MQTT packet for. The encoded length must be greater than [MqttFixedHeader.MAX_REMAINING_LEN](#) (=268435455 bytes) in order to cause this error.

Shorten the number of topics or the length of the topic strings.

Parameters

- **packet_id** (*int*) – $0 \leq \text{packet_id} \leq 2^{16}-1$
- **topics** (*iterable of str*) –

classmethod `decode_body(header, f)`

Generates a *MqttUnsubscribe* packet given a *MqttFixedHeader*. This method asserts that `header.packet_type` is *unsubscribe*.

Parameters

- **header** (*MqttFixedHeader*) –
- **f** (*file*) – Object with a read method.

Raises *DecodeError* – When there are extra bytes at the end of the packet.

Returns

- *int* – Number of bytes consumed from *f*.
- *MqttUnsubscribe* – Object extracted from *f*.

encode_body (*f*)

Parameters **f** (*file*) – File-like object with write method.

Returns Number of bytes written to file.

Return type *int*

packet_id

int – packet id such that $0 \leq \text{packet_id} \leq 2^{16}-1$.

topics

tuple of str – Topics to be unsubscribed.

class `mqtt_codec.packet.MqttWill(qos, topic, message, retain)`

Bases: *object*

An immutable class representing an MQTT Will message as described beginning in [MQTT-3.1.2-8, line 471].

Parameters

- **qos** (*int*) – $0 \leq \text{qos} \leq 2$
- **topic** (*str*) –
- **message** (*bytes*) –
- **retain** (*bool*) –

message

bytes – Will message.

qos

int – A number such that $0 \leq \text{self.qos} \leq 2$.

retain

bool – Will retain flag as described in MQTT spec line 504 section 3.1.2.7. In general, with the retain flag set the will message will be saved and published to clients as they connect to the server.

topic

str – Topic name.

class `mqtt_codec.packet.SubscribeResult`

Bases: *enum.IntEnum*

fail = 128

qos0 = 0

qos1 = 1

`qos2 = 2`

`mqtt_codec.packet.are_flags_valid(packet_type, flags)`

True when flags comply with [MQTT-2.2.2-1] requirements based on `packet_type`; False otherwise.

Parameters

- **packet_type** (`MqttControlPacketType`) –
- **flags** (`int`) – Integer representation of 4-bit MQTT header flags field. Values outside of the range [0, 15] will certainly cause the function to return False.

Returns

Return type `bool`

3.2.2 `mqtt_codec.io` Package

A collection of helper functions and classes used to read MQTT control packets.

class `mqtt_codec.io.BytesReader(buf)`

Bases: `object`

Creates a file-like object that reads from a buffer.

Parameters `buf` (`bytes` or `bytearray`) – Object to read from.

close()

Read operations conducted after this method is called will raise `ValueError`. This makes the object behave like other read objects even though no resources are freed.

closed

`bool` – True if `self.close()` has been called; False otherwise.

read(max_bytes=1)

Read at most `max_bytes` from internal buffer.

Parameters `max_bytes` (`int`) – Maximum number of bytes to read.

Raises `ValueError` – If read is called after close has been called.

Returns Bytes extracted from internal buffer. Length may be less than `max_bytes`. On end-of file returns a bytes object with zero-length.

Return type `bytes`

exception `mqtt_codec.io.DecodeError`

Bases: `exceptions.Exception`

exception `mqtt_codec.io.EncodeError`

Bases: `exceptions.Exception`

class `mqtt_codec.io.FileDecoder(f)`

Bases: `object`

Creates an object that extracts values from the file-like object `f`.

Parameters `f` (`file`) – Object with read method.

num_bytes_consumed

`int` – number of bytes consumed from underlying stream.

read(num_bytes)

Read `num_bytes` and return them.

Parameters `num_bytes` (*int*) – Number of bytes to extract from the underlying stream.

Raises `UnderflowDecodeError` – Raised when a read failed to extract enough bytes from the underlying stream to extract the bytes.

Returns A bytes object extracted from underlying stream.

Return type bytes

unpack (*struct*)

Read as many bytes as are required to extract struct then unpack and return a tuple of the values.

Raises `UnderflowDecodeError` – Raised when a read failed to extract enough bytes from the underlying stream to extract the bytes.

Parameters `struct` (*struct.Struct*) –

Returns Tuple of extracted values.

Return type tuple

unpack_bytes ()

Unpack a utf-8 string encoded as described in MQTT Version 3.1.1 section 1.5.3 line 177. This is a 16-bit unsigned length followed by a utf-8 encoded string.

Returns

- *int* – Number of bytes consumed
- *bytes* – A bytes object extracted from the underlying stream.

unpack_utf8 ()

Decode a utf-8 string encoded as described in MQTT Version 3.1.1 section 1.5.3 line 177. This is a 16-bit unsigned length followed by a utf-8 encoded string.

Raises

- `UnderflowDecodeError` – Raised when a read failed to extract enough bytes from the underlying stream to decode the string.
- `DecodeError` – When any code point in the utf-8 string is invalid.

Returns

- *int* – Number of bytes consumed.
- *str* – A string utf-8 decoded from the underlying stream.

unpack_varint (*max_bytes*)

Decode variable integer using algorithm similar to that described in MQTT Version 3.1.1 line 297.

Parameters `max_bytes` (*int or None*) – If a varint cannot be constructed using *max_bytes* or fewer from *f* then raises a `DecodeError`. If *None* then there is no maximum number of bytes.

Raises

- `DecodeError` – When length is greater than *max_bytes*.
- `UnderflowDecodeError` – When file ends before enough bytes can be read to construct the varint.

Returns

- *int* – Number of bytes consumed.
- *int* – Value extracted from *f*.

class mqtt_codec.io.LimitReader (*f*, *limit=None*)

Bases: object

Reads up to `limit` bytes from the underlying file. If `limit` is none then reads to the end of the file.

Parameters

- **f** (*file*) – File-like object with read method.
- **limit** (*int optional*) – Maximum number of bytes to read from the underlying file or None the reader should continue until the end of the file.

limit

int or None – maximum number of bytes to read from underlying stream.

read (*max_bytes=1*)

Read at most *max_bytes* from internal buffer.

Parameters **max_bytes** (*int*) – Maximum number of bytes to read.

Returns Bytes extracted from internal buffer. Length may be less than *max_bytes*. On end-of file returns a bytes object with zero-length.

Return type bytes

exception mqtt_codec.io.OverflowEncodeError

Bases: *mqtt_codec.io.EncodeError*

exception mqtt_codec.io.OversizePacketEncodeError

Bases: *mqtt_codec.io.EncodeError*

Raised when the parameters used to create the MQTT packet would result in an impossibly large packet.

exception mqtt_codec.io.UnderflowDecodeError

Bases: *mqtt_codec.io.DecodeError*

exception mqtt_codec.io.Utf8DecodeError (*e*)

Bases: *mqtt_codec.io.DecodeError*

mqtt_codec.io.decode_bytes (*f*)

Decode a buffer length from a 2-byte unsigned int then read the subsequent bytes.

Parameters **f** (*file*) – File-like object with read method.

Raises *UnderflowDecodeError* – When the end of stream is encountered before the end of the encoded bytes.

Returns

- *int* – Number of bytes read from *f*.
- *bytes* – Value bytes decoded from *f*.

mqtt_codec.io.decode_utf8 (*f*)

Decode a utf-8 string encoded as described in MQTT Version 3.1.1 section 1.5.3 line 177. This is a 16-bit unsigned length followed by a utf-8 encoded string.

Parameters **f** (*file*) – File-like object with read method.

Raises

- *UnderflowDecodeError* – Raised when a read failed to extract enough bytes from the underlying stream to decode the string.
- *Utf8DecodeError* – When any code point in the utf-8 string is invalid.

Returns

- *int* – Number of bytes consumed.
- *str* – A string utf-8 decoded from *f*.

`mqtt_codec.io.decode_varint(f, max_bytes=4)`

Decode variable integer using algorithm similar to that described in MQTT Version 3.1.1 line 297.

Parameters

- **f** (*file*) – Object with a read method.
- **max_bytes** (*int* or *None*) – If a varint cannot be constructed using *max_bytes* or fewer from *f* then raises a *DecodeError*. If *None* then there is no maximum number of bytes.

Raises

- *DecodeError* – When length is greater than *max_bytes*.
- *UnderflowDecodeError* – When file ends before enough bytes can be read to construct the varint.

Returns

- *int* – Number of bytes consumed.
- *int* – Value extracted from *f*.

`mqtt_codec.io.encode_bytes(src_buf, dst_file)`

Encode a buffer length followed by the bytes of the buffer itself.

Parameters

- **src_buf** (*bytes*) – Source bytes to be encoded. Function asserts that $0 \leq \text{len}(\text{src_buf}) \leq 2^{16}-1$.
- **dst_file** (*file*) – File-like object with write method.

Returns Number of bytes written to *dst_file*.

Return type *int*

`mqtt_codec.io.encode_utf8(s, f)`

UTF-8 encodes string *s* to file-like object *f* according to the MQTT Version 3.1.1 specification in section 1.5.3.

The maximum length for the encoded string is $2^{16}-1$ (65535) bytes. An assertion error will result if the encoded string is longer.

Parameters

- **s** (*str*) – String to be encoded.
- **f** (*file*) – File-like object.

Returns Number of bytes written to *f*.

Return type *int*

`mqtt_codec.io.encode_varint(v, f)`

Encode integer *v* to file *f*.

Parameters

- **v** (*int*) – Integer $v \geq 0$.
- **f** (*file*) – Object containing a write method.

Returns Number of bytes written.

Return type int

3.3 Change Log

3.3.1 1.0.2 (2019-01-27)

Fix

#5: QoS=0 publish messages incorrectly read/write packet_id.

In Publish messages with QoS=0, packet_id is being serialized and deserialized in violation of MQTT 3.1.1 spec (See 3.3.2.2).

<https://github.com/kcallin/haka-mqtt/issues/5>

#6: Corrupt MqttSuback results in non-DecodeError exception.

While decoding MqttSuback, a corrupted SubscribeResult results in a TypeError instead of DecodeError. This violates the decode method's interface spec.

<https://github.com/kcallin/mqtt-codec/issues/6>

3.3.2 1.0.1 (2018-11-28)

New

#4: MqttConnect.__repr__ has seconds units on keep_alive.

<https://github.com/kcallin/haka-mqtt/issues/21>

3.3.3 1.0.0 (2018-11-24)

New

- First stable production release.

3.3.4 0.1.3 (2018-11-17)

New

- Python 3 support.
- Updating packaging mechanism.

3.3.5 0.1.2 (2018-11-15)

New

- Python 3 compatibility.
- MqttConnect object now has read-only attributes.

Fixes

#2: MqttPublish.payload sometimes has a type that is not bytes.

<https://github.com/kcallin/mqtt-codec/issues/2>

3.3.6 0.1.1 (2018-10-22)

- Documentation improvements.
- MqttConnect.__str__ no longer shows user/pass.
- setup.py:install_requires now compatible with setuptools 18.

3.3.7 0.1.0 (2018-10-03)

Initial release.

3.4 Developer Guide

The developer's guide is for a person who wants to change and contribute changes to *mqtt-codec*. It builds on information in *User Guide*.

3.4.1 Uncontrolled Builds

Uncontrolled source builds are created in the standard python fashion:

```
$ python setup.py sdist
running sdist
running egg_info
writing requirements to mqtt_codec.egg-info/requirements.txt
writing mqtt_codec.egg-info/PKG-INFO
writing top-level names to mqtt_codec.egg-info/top_level.txt
writing dependency_links to mqtt_codec.egg-info/dependency_links.txt
reading manifest file 'mqtt_codec.egg-info/SOURCES.txt'
writing manifest file 'mqtt_codec.egg-info/SOURCES.txt'
running check
creating mqtt-codec-0.1.0-uncontrolled-20180907
creating mqtt-codec-0.1.0-uncontrolled-20180907/mqtt_codec
creating mqtt-codec-0.1.0-uncontrolled-20180907/mqtt_codec.egg-info
[... removed for brevity ...]
copying tests/test_reactor.py -> mqtt-codec-0.1.0-uncontrolled-20180907/tests
copying tests/test_scheduler.py -> mqtt-codec-0.1.0-uncontrolled-20180907/tests
Writing mqtt-codec-0.1.0-uncontrolled-20180907/setup.cfg
creating dist
Creating tar archive
removing 'mqtt-codec-0.1.0-uncontrolled-20180907' (and everything under it)
$ ls dist
mqtt-codec-0.1.0-uncontrolled-20180907.tar.gz
$
```

The output artifact has the word “uncontrolled” along with a build date so that users will know the artifact is not a release or from a continuous integration build server.

3.4.2 Tests

The *mqtt-codec* library comes with a battery of tests.

The built-in automated tests can be run from the command-line using `setup.py`.

```
$ python setup.py test
$
```

3.4.3 Coverage

Test coverage is monitored using [coverage.py](#) version 4.5 or higher. Normally this can be installed through your operating system's package manager (like `rpm` or `apt-get`) or by using *pip*. A coverage configuration file is included at `.coveragerc` and the tool can be run in this fashion:

```
$ coverage run setup.py test
running test
running egg_info
writing requirements to mqtt_codec.egg-info/requirements.txt
writing mqtt_codec.egg-info/PKG-INFO
writing top-level names to mqtt_codec.egg-info/top_level.txt
writing dependency_links to mqtt_codec.egg-info/dependency_links.txt
reading manifest file 'mqtt_codec.egg-info/SOURCES.txt'
writing manifest file 'mqtt_codec.egg-info/SOURCES.txt'
running build_ext
test_read_after_close (tests.test_io.TestBytesReader) ... ok
test_body_underflow (tests.test_io.TestDecodeBytes) ... ok
[... removed for brevity...]
test_subscribe (tests.test_mqtt.TestUnsubscribe) ... ok
test_decode_encode (tests.test_mqtt.TestUtf8Codec) ... ok
test_encode_max_len_utf8 (tests.test_mqtt.TestUtf8Codec) ... ok
test_encode_too_long_utf8 (tests.test_mqtt.TestUtf8Codec) ... ok
```

```
-----
Ran 48 tests in 0.014s
```

```
OK
```

```
$ coverage report
```

Name	Stmts	Miss	Branch	BrPart	Cover
mqtt_codec/__init__.py	0	0	0	0	100%
mqtt_codec/io.py	162	4	32	1	97%
mqtt_codec/packet.py	587	40	110	27	89%
TOTAL	749	44	142	28	91%

3.4.4 Docstrings

Python source code is documented according to the the numpy documentation standard at <https://numpydoc.readthedocs.io/en/latest/format.html>.

3.4.5 Documentation

The documentation for `mqtt-codec` is created with [Sphinx](#) and is build the fashion usual to that framework:

```
$ cd doc
$ make html
$
```

The documentation contains doctests which can be verified in this fashion:

```
$ make doctest
Running Sphinx v1.7.7
loading pickled environment... done
building [mo]: targets for 0 po files that are out of date
building [doctest]: targets for 5 source files that are out of date
updating environment: 0 added, 1 changed, 0 removed
reading sources... [100%] user_guide
looking for now-outdated files... none found
pickling environment... done
checking consistency... done
running tests...

Document: user_guide
-----
1 items passed all tests:
14 tests in default
14 tests in 1 items.
14 passed and 0 failed.
Test passed.

Doctest summary
=====
    14 tests
    0 failures in tests
    0 failures in setup code
    0 failures in cleanup code
build succeeded.

Testing of doctests in the sources finished, look at the results in build/doctest/
↪output.txt.
```

As suggested by the text, the output can be found in `build/doctest/output.txt`.

3.4.6 Contributing Changes

If you have an idea for an enhancement then you are welcome to fork the github repository at <https://github.com/kcallin/mqtt-codec>, make your changes, and then submit a pull request.

Minimally, your git commit record must have the following:

1. Your name and e-mail address captured in the “Author” field.
2. A single line summary in the message field followed by a more detailed description.
3. A “Signed-off-by” entry with matching credentials in the message footer.

If the commit fixes a bug then a link should be included in the message footer. The id (bug number) of the bug should also be included in the message summary.

You can specify additional authors using one or more “Also-by” entries in the message footer.

For example:

```
commit 862e6ff22ad56c10df6de3385ffa4c7d02363d1d
Author: Joe Somebody <somebody@someplace.net>
Date: Mon Jun 17 17:19:38 2013 -0700

[2] MqttPublish.payload must be bytes

The MqttPublish payload parameter is stored and returned by
MqttPublish.payload without checking that it is bytes. This
change adds an assertion that the payload parameter is bytes.

Bug: https://github.com/kcallin/mqtt-codec/issues/3
Also-by: Some Otherperson <otherperson@someplace.net>
Signed-off-by: Joe Somebody <somebody@someplace.net>
```

The “Signed-off-by” entry is required. By including this, you confirm that you are in compliance with the [Certificate of Origin](#).

Note that the footer entries must occur at the bottom of the commit message and must not include any blank lines.

Signing off on a Commit

Git contains built-in support for signing off on a commit.

From command-line *git*, add -s to the command:

```
$ git commit -s --gpg-sign[=<keyid>] -m "[2] MqttPublish.payload must be bytes"
```

3.5 Distribution Procedure

The release procedure was created using information from these core sources:

- [PEP 503 - Simple Repository API](#)
- [Python Packaging User Guide](#)
- [Twine](#)

3.5.1 Test Release

Verify that version and release numbers in `doc/source/conf.py` match `setup.py`.

```
$ grep -e version -e release doc/source/conf.py
# The short X.Y version
version = u'1.0.0'
# The full version, including alpha/beta/rc tags
release = u'1.0.0'
$
```

Ensure there are no old build artifacts.

```
$ rm -rf dist build mqtt_codec.egg-info htmlcov
$ ls dist
$
```

It's a common problem to accidentally forget to commit important changes. To combat this the `pyvertest.py` procedure clones the `mqtt-codec` repository, passes it to a docker container, and runs a test battery in a set of environments.

```
$ ./pyvertest.py
[... removed for brevity ...]
pip install python:3.7-alpine3.8
docker run --rm -v /home/kcallin/src/mqtt-codec:/mqtt-codec python:3.7-alpine3.8 pip_
↳ install /mqtt-codec
Processing /mqtt-codec
Building wheels for collected packages: mqtt-codec
  Running setup.py bdist_wheel for mqtt-codec: started
  Running setup.py bdist_wheel for mqtt-codec: finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/c1/64/0f/
  ↳ d02b6f3717526372cf5d4a5beb9b63181eb54bd4ed964fa7e1
Successfully built mqtt-codec
Installing collected packages: mqtt-codec
Successfully installed mqtt-codec-1.0.0-uncontrolled-20181125
Return code 0
> All okay.
```

Ensure that `CHANGELOG.rst` has release version and release date correct as well as release content listed.

```
$ vi CHANGELOG.rst
$ git commit -S CHANGELOG.rst
```

Create test release artifacts.

```
$ python setup.py egg_info -D -b 'a' sdist
running sdist
running egg_info
writing requirements to mqtt_codec.egg-info/requirements.txt
writing mqtt_codec.egg-info/PKG-INFO
writing top-level names to mqtt_codec.egg-info/top_level.txt
writing dependency_links to mqtt_codec.egg-info/dependency_links.txt
reading manifest file 'mqtt_codec.egg-info/SOURCES.txt'
writing manifest file 'mqtt_codec.egg-info/SOURCES.txt'
running check
creating mqtt-codec-0.1.2
creating mqtt-codec-0.1.2/mqtt_codec
[... removed for brevity ...]
copying tests/test_reactor.py -> mqtt-codec-0.1.2/tests
copying tests/test_scheduler.py -> mqtt-codec-0.1.2/tests
Writing mqtt-codec-0.1.2/setup.cfg
Creating tar archive
removing 'mqtt-codec-0.1.2' (and everything under it)
$ ls dist
mqtt-codec-0.1.2.tar.gz
$
```

GPG sign test release artifact:

```
$ gpg --detach-sign -a dist/mqtt-codec-0.1.2.tar.gz

You need a passphrase to unlock the secret key for
user: "Keegan Callin <kc@kcallin.net>"
4096-bit RSA key, ID DD53792F, created 2017-01-01 (main key ID 14BC2EFF)
```

(continues on next page)

(continued from previous page)

```
gpg: gpg-agent is not available in this session
$ ls dist
mqtt-codec-0.1.2.tar.gz  mqtt-codec-0.1.2.tar.gz.asc
$ gpg --verify dist/mqtt-codec-0.1.2.tar.gz.asc
gpg: assuming signed data in `dist/mqtt-codec-0.1.2.tar.gz'
gpg: Signature made Sat 01 Sep 2018 11:00:31 AM MDT using RSA key ID DD53792F
gpg: Good signature from "Keegan Callin <kc@kcallin.net>" [ultimate]
Primary key fingerprint: BD51 01F1 9699 A719 E563 6D85 4A4A 7B98 14BC 2EFF
Subkey fingerprint: BE56 D781 0163 488F C7AE 62AC 3914 0AE2 DD53 792F
$
```

Ensure that twine version 1.12.0 or high is installed:

```
$ twine --version
twine version 1.12.0 (pkginfo: 1.4.2, requests: 2.20.1, setuptools: 40.6.2,
requests-toolbelt: 0.8.0, tqdm: 4.28.1)
```

Verify that distribution passes twine checks:

```
$ twine check dist/*
Checking distribution dist/mqtt-codec-1.0.1.tar.gz: Passed
```

Release artifacts to **TEST** PyPI.

```
$ twine upload --repository-url https://test.pypi.org/legacy/ dist/*
Uploading distributions to https://test.pypi.org/legacy/
Enter your username: kc
Enter your password:
Uploading mqtt-codec-0.1.2.tar.gz
$
```

The resulting [TestPyPI entry](#) should be inspected for correctness. “The database for TestPyPI may be periodically pruned, so it is not unusual for user accounts to be deleted¹”. Packages on **TEST** PyPI and **real** PyPI cannot be removed upon distributor demand. On **TEST** PyPI packages may be removed on prune, on **real** PyPI they will remain forever. A checklist to help verify the PyPI is:

- Version Number is Correct
- Documentation Link is Correct
- ReST README.rst is rendered correctly on the front page.

After the checklist is complete then it is time to upload to **real** PyPI and verify that the release is complete. There is no undoing this operation. Think Carefully.

PEP 508 – Dependency specification for Python Software Packages

PEP-314 – Metadata for Python Software Packages v1.1

3.5.2 Official Release

Create, sign, and push release tag:

```
$ git tag -s v0.1.0
$ git push origin v0.1.0
```

¹ [Test PyPI, Registering Your Account](#), retrieved 2018-09-07.

Remove test artifacts:

```
$ rm -rf dist build mqtt_codec.egg-info htmlcov
$ ls dist
$
```

Create official release artifacts.

```
$ python setup.py egg_info -D -b '' sdist
running sdist
running egg_info
writing requirements to mqtt_codec.egg-info/requirements.txt
writing mqtt_codec.egg-info/PKG-INFO
writing top-level names to mqtt_codec.egg-info/top_level.txt
writing dependency_links to mqtt_codec.egg-info/dependency_links.txt
reading manifest file 'mqtt_codec.egg-info/SOURCES.txt'
writing manifest file 'mqtt_codec.egg-info/SOURCES.txt'
running check
creating mqtt-codec-0.1.2
creating mqtt-codec-0.1.2/mqtt_codec
[... removed for brevity ...]
copying tests/test_reactor.py -> mqtt-codec-0.1.2/tests
copying tests/test_scheduler.py -> mqtt-codec-0.1.2/tests
Writing mqtt-codec-0.1.2/setup.cfg
Creating tar archive
removing 'mqtt-codec-0.1.2' (and everything under it)
$ ls dist
mqtt-codec-0.1.2.tar.gz
$
```

GPG sign official release artifact:

```
$ gpg --detach-sign -a dist/mqtt-codec-0.1.2.tar.gz

You need a passphrase to unlock the secret key for
user: "Keegan Callin <kc@kcallin.net>"
4096-bit RSA key, ID DD53792F, created 2017-01-01 (main key ID 14BC2EFF)

gpg: gpg-agent is not available in this session
$ ls dist
mqtt-codec-0.1.2.tar.gz  mqtt-codec-0.1.2.tar.gz.asc
$ gpg --verify dist/mqtt-codec-0.1.2.tar.gz.asc
gpg: assuming signed data in `dist/mqtt-codec-0.1.2.tar.gz'
gpg: Signature made Sat 01 Sep 2018 11:00:31 AM MDT using RSA key ID DD53792F
gpg: Good signature from "Keegan Callin <kc@kcallin.net>" [ultimate]
Primary key fingerprint: BD51 01F1 9699 A719 E563 6D85 4A4A 7B98 14BC 2EFF
Subkey fingerprint: BE56 D781 0163 488F C7AE 62AC 3914 0AE2 DD53 792F
$
```

The access credentials in `~/.pypirc` contains the username/password that twine uses for PyPI.

```
$ cat ~/.pypirc
[distutils]
index-servers =
    pypi

[pypi]
username:<XXXXXX>
```

(continues on next page)

(continued from previous page)

```
password:<XXXXXX>
$ twine upload dist/*
```

3.5.3 Distribute Documentation

Documentation is distributed through readthedocs.org. After a release visit the [mqtt-codec readthedocs Version](#), page and make sure the correct versions are marked as “Active”.

The `mqtt-codec` project documentation uses [PlantUML](#) to draw diagrams and this package is not support out-of-the-box by *readthedocs*. The project root directory contains a `.readthedocs.yml` file to set the build *readthedocs* build environment to one that supports PlantUML and bypass the problem.

3.5.4 Increment Version Number

The release number in *setup.py* has been consumed and should never be used again. Take the time to increment the number, commit the change, then push the change.

```
$ vi setup.py
$ vi doc/source/conf.py
$ git commit setup.py
$ git push origin master
```


CHAPTER 4

Indices and tables

- `genindex`
- `modindex`
- `search`

m

- `mqtt_codec`, 9
- `mqtt_codec.io`, 24
- `mqtt_codec.packet`, 9

A

accepted (mqtt_codec.packet.ConnackResult attribute), 11

are_flags_valid() (in module mqtt_codec.packet), 24

B

BytesReader (class in mqtt_codec.io), 24

C

clean_session (mqtt_codec.packet.MqttConnect attribute), 12

client_id (mqtt_codec.packet.MqttConnect attribute), 12

close() (mqtt_codec.io.BytesReader method), 24

closed (mqtt_codec.io.BytesReader attribute), 24

connack (mqtt_codec.packet.MqttControlPacketType attribute), 13

ConnackResult (class in mqtt_codec.packet), 9

connect (mqtt_codec.packet.MqttControlPacketType attribute), 13

CONNECT_HEADER (mqtt_codec.packet.MqttConnect attribute), 12

D

decode() (mqtt_codec.packet.MqttFixedHeader static method), 14

decode() (mqtt_codec.packet.MqttPacketBody class method), 15

decode_body() (mqtt_codec.packet.MqttConnack class method), 11

decode_body() (mqtt_codec.packet.MqttConnect class method), 12

decode_body() (mqtt_codec.packet.MqttDisconnect class method), 13

decode_body() (mqtt_codec.packet.MqttPacketBody class method), 15

decode_body() (mqtt_codec.packet.MqttPingreq class method), 15

decode_body() (mqtt_codec.packet.MqttPingresp class method), 16

decode_body() (mqtt_codec.packet.MqttPuback class method), 16

decode_body() (mqtt_codec.packet.MqttPubcomp class method), 17

decode_body() (mqtt_codec.packet.MqttPublish class method), 18

decode_body() (mqtt_codec.packet.MqttPubrec class method), 19

decode_body() (mqtt_codec.packet.MqttPubrel class method), 20

decode_body() (mqtt_codec.packet.MqttSuback class method), 20

decode_body() (mqtt_codec.packet.MqttSubscribe class method), 21

decode_body() (mqtt_codec.packet.MqttUnsuback class method), 22

decode_body() (mqtt_codec.packet.MqttUnsubscribe class method), 22

decode_bytes() (in module mqtt_codec.io), 26

decode_utf8() (in module mqtt_codec.io), 26

decode_varint() (in module mqtt_codec.io), 27

DecodeError, 24

disconnect (mqtt_codec.packet.MqttControlPacketType attribute), 13

dupe (mqtt_codec.packet.MqttPublish attribute), 18, 19

E

encode() (mqtt_codec.packet.MqttFixedHeader method), 14

encode() (mqtt_codec.packet.MqttPacketBody method), 15

encode_body() (mqtt_codec.packet.MqttConnack method), 11

encode_body() (mqtt_codec.packet.MqttConnect method), 12

encode_body() (mqtt_codec.packet.MqttDisconnect method), 13

encode_body() (mqtt_codec.packet.MqttPacketBody method), 15

[encode_body\(\)](#) ([mqtt_codec.packet.MqttPingreq](#) method), [15](#)
[encode_body\(\)](#) ([mqtt_codec.packet.MqttPingresp](#) method), [16](#)
[encode_body\(\)](#) ([mqtt_codec.packet.MqttPuback](#) method), [16](#)
[encode_body\(\)](#) ([mqtt_codec.packet.MqttPubcomp](#) method), [17](#)
[encode_body\(\)](#) ([mqtt_codec.packet.MqttPublish](#) method), [19](#)
[encode_body\(\)](#) ([mqtt_codec.packet.MqttPubrec](#) method), [19](#)
[encode_body\(\)](#) ([mqtt_codec.packet.MqttPubrel](#) method), [20](#)
[encode_body\(\)](#) ([mqtt_codec.packet.MqttSuback](#) method), [21](#)
[encode_body\(\)](#) ([mqtt_codec.packet.MqttSubscribe](#) method), [21](#)
[encode_body\(\)](#) ([mqtt_codec.packet.MqttUnsuback](#) method), [22](#)
[encode_body\(\)](#) ([mqtt_codec.packet.MqttUnsubscribe](#) method), [23](#)
[encode_bytes\(\)](#) (in module [mqtt_codec.io](#)), [27](#)
[encode_utf8\(\)](#) (in module [mqtt_codec.io](#)), [27](#)
[encode_varint\(\)](#) (in module [mqtt_codec.io](#)), [27](#)
[EncodeError](#), [24](#)

F

[fail](#) ([mqtt_codec.packet.SubscribeResult](#) attribute), [23](#)
[fail_bad_client_id](#) ([mqtt_codec.packet.ConnackResult](#) attribute), [11](#)
[fail_bad_protocol_version](#) ([mqtt_codec.packet.ConnackResult](#) attribute), [11](#)
[fail_bad_username_or_password](#) ([mqtt_codec.packet.ConnackResult](#) attribute), [11](#)
[fail_not_authorized](#) ([mqtt_codec.packet.ConnackResult](#) attribute), [11](#)
[fail_server_unavailable](#) ([mqtt_codec.packet.ConnackResult](#) attribute), [11](#)
[FileDecoder](#) (class in [mqtt_codec.io](#)), [24](#)
[flags](#) ([mqtt_codec.packet.MqttFixedHeader](#) attribute), [14](#)

K

[keep_alive](#) ([mqtt_codec.packet.MqttConnect](#) attribute), [12](#)

L

[limit](#) ([mqtt_codec.io.LimitReader](#) attribute), [26](#)
[LimitReader](#) (class in [mqtt_codec.io](#)), [25](#)

M

[max_qos](#) ([mqtt_codec.packet.MqttTopic](#) attribute), [22](#)

[MAX_REMAINING_LEN](#) ([mqtt_codec.packet.MqttFixedHeader](#) attribute), [14](#)
[message](#) ([mqtt_codec.packet.MqttWill](#) attribute), [23](#)
[mqtt_codec](#) (module), [9](#)
[mqtt_codec.io](#) (module), [24](#)
[mqtt_codec.packet](#) (module), [9](#)
[MqttConnack](#) (class in [mqtt_codec.packet](#)), [11](#)
[MqttConnect](#) (class in [mqtt_codec.packet](#)), [11](#)
[MqttControlPacketType](#) (class in [mqtt_codec.packet](#)), [12](#)
[MqttDisconnect](#) (class in [mqtt_codec.packet](#)), [13](#)
[MqttFixedHeader](#) (class in [mqtt_codec.packet](#)), [13](#)
[MqttPacketBody](#) (class in [mqtt_codec.packet](#)), [15](#)
[MqttPingreq](#) (class in [mqtt_codec.packet](#)), [15](#)
[MqttPingresp](#) (class in [mqtt_codec.packet](#)), [15](#)
[MqttPuback](#) (class in [mqtt_codec.packet](#)), [16](#)
[MqttPubcomp](#) (class in [mqtt_codec.packet](#)), [16](#)
[MqttPublish](#) (class in [mqtt_codec.packet](#)), [17](#)
[MqttPubrec](#) (class in [mqtt_codec.packet](#)), [19](#)
[MqttPubrel](#) (class in [mqtt_codec.packet](#)), [20](#)
[MqttSuback](#) (class in [mqtt_codec.packet](#)), [20](#)
[MqttSubscribe](#) (class in [mqtt_codec.packet](#)), [21](#)
[MqttTopic](#) (class in [mqtt_codec.packet](#)), [21](#)
[MqttUnsuback](#) (class in [mqtt_codec.packet](#)), [22](#)
[MqttUnsubscribe](#) (class in [mqtt_codec.packet](#)), [22](#)
[MqttWill](#) (class in [mqtt_codec.packet](#)), [23](#)

N

[name](#) ([mqtt_codec.packet.MqttTopic](#) attribute), [22](#)
[num_bytes_consumed](#) ([mqtt_codec.io.FileDecoder](#) attribute), [24](#)

O

[OverflowEncodeError](#), [26](#)
[OversizePacketEncodeError](#), [26](#)

P

[packet\(\)](#) ([mqtt_codec.packet.MqttFixedHeader](#) method), [14](#)
[packet_id](#) ([mqtt_codec.packet.MqttPuback](#) attribute), [16](#)
[packet_id](#) ([mqtt_codec.packet.MqttPubcomp](#) attribute), [17](#)
[packet_id](#) ([mqtt_codec.packet.MqttPublish](#) attribute), [18](#), [19](#)
[packet_id](#) ([mqtt_codec.packet.MqttPubrec](#) attribute), [19](#)
[packet_id](#) ([mqtt_codec.packet.MqttPubrel](#) attribute), [20](#)
[packet_id](#) ([mqtt_codec.packet.MqttSuback](#) attribute), [21](#)
[packet_id](#) ([mqtt_codec.packet.MqttSubscribe](#) attribute), [21](#)
[packet_id](#) ([mqtt_codec.packet.MqttUnsuback](#) attribute), [22](#)
[packet_id](#) ([mqtt_codec.packet.MqttUnsubscribe](#) attribute), [23](#)

packet_type (mqtt_codec.packet.MqttFixedHeader attribute), 14

password (mqtt_codec.packet.MqttConnect attribute), 12

payload (mqtt_codec.packet.MqttPublish attribute), 18, 19

pingreq (mqtt_codec.packet.MqttControlPacketType attribute), 13

pingresp (mqtt_codec.packet.MqttControlPacketType attribute), 13

PROTOCOL_LEVEL (mqtt_codec.packet.MqttConnect attribute), 12

puback (mqtt_codec.packet.MqttControlPacketType attribute), 13

pubcomp (mqtt_codec.packet.MqttControlPacketType attribute), 13

publish (mqtt_codec.packet.MqttControlPacketType attribute), 13

pubrec (mqtt_codec.packet.MqttControlPacketType attribute), 13

pubrel (mqtt_codec.packet.MqttControlPacketType attribute), 13

Q

qos (mqtt_codec.packet.MqttPublish attribute), 18, 19

qos (mqtt_codec.packet.MqttWill attribute), 23

qos0 (mqtt_codec.packet.SubscribeResult attribute), 23

qos1 (mqtt_codec.packet.SubscribeResult attribute), 23

qos2 (mqtt_codec.packet.SubscribeResult attribute), 23

R

read() (mqtt_codec.io.BytesReader method), 24

read() (mqtt_codec.io.FileDecoder method), 24

read() (mqtt_codec.io.LimitReader method), 26

remaining_len (mqtt_codec.packet.MqttFixedHeader attribute), 14

results (mqtt_codec.packet.MqttSuback attribute), 21

retain (mqtt_codec.packet.MqttPublish attribute), 18, 19

retain (mqtt_codec.packet.MqttWill attribute), 23

return_code (mqtt_codec.packet.MqttConnack attribute), 11

S

session_present (mqtt_codec.packet.MqttConnack attribute), 11

size (mqtt_codec.packet.MqttFixedHeader attribute), 14

suback (mqtt_codec.packet.MqttControlPacketType attribute), 13

subscribe (mqtt_codec.packet.MqttControlPacketType attribute), 13

SubscribeResult (class in mqtt_codec.packet), 23

T

topic (mqtt_codec.packet.MqttPublish attribute), 18, 19

topic (mqtt_codec.packet.MqttWill attribute), 23

topics (mqtt_codec.packet.MqttSubscribe attribute), 21

topics (mqtt_codec.packet.MqttUnsubscribe attribute), 23

U

UnderflowDecodeError, 26

unpack() (mqtt_codec.io.FileDecoder method), 25

unpack_bytes() (mqtt_codec.io.FileDecoder method), 25

unpack_utf8() (mqtt_codec.io.FileDecoder method), 25

unpack_varint() (mqtt_codec.io.FileDecoder method), 25

unsuback (mqtt_codec.packet.MqttControlPacketType attribute), 13

unsubscribe (mqtt_codec.packet.MqttControlPacketType attribute), 13

username (mqtt_codec.packet.MqttConnect attribute), 12

Utf8DecodeError, 26

W

will (mqtt_codec.packet.MqttConnect attribute), 12