

MOQT QLOG

[draft-pardue-moq-qlog-moq-events](#)

Lucas Pardue, Mathis Engelbart

IETF 122, 2025-03

What is qlog again?

Structured logging for network protocols

QUIC WG is working on 3 schemas:

[Main schema](#): extensible framework, common events, file formats, etc

[QUIC schema](#): events for QUIC connections, streams, packets, frames etc

[HTTP/3 schema](#): events for HTTP/3 streams, requests, responses, etc

Schema defined in **CDDL**

Serialization format agnostic .. but we provide two out-of-the box formats

Regular JSON

Streaming JSON

Main schema abstract event

```
Event = {  
  time: float64  
  name: text  
  data: $ProtocolEventData  
  ? path: PathID  
  ? time_format: TimeFormat  
  ? group_id: GroupID  
  ? system_info: SystemInformation  
  
  ; events can contain any amount  
  of custom fields  
  * text => any  
}
```

QUIC schema concrete event

```
QUICPacketSent = {  
  header: PacketHeader  
  ? frames: [* $QuicFrame]  
  <... more fields>  
}
```

Example JSON

```
{  
  "time":17.956148,"name":"transport:packet_sent","data":{"header":{"packet_type":  
"1RTT","packet_number":3},"raw":{"length":118,"payload_length":80},"frames":  
[{"frame_type":"stream","stream_id":0,"offset":0,"length":75,"fin":true}]}}
```

MOQT QLOG schema

Top-level event

```
MOQTControlMessageCreated = {  
  stream_id: uint64  
  ? length: uint64  
  message: $MOQTControlMessage  
  ? raw: RawInfo  
  * $$moqt-controlmessagecreated-extension  
}
```

Message concrete types

```
MOQTClientSetupMessage = {  
  type: "client setup"  
  number_of_supported_versions: uint64  
  supported_versions: [* uint64]  
  number_of_parameters: uint64  
  ? setup_parameters: [* $MOQTSetupParameter]  
}
```

```
MOQTServerSetupMessage = {  
  type: "server setup"  
  selected_version: uint64  
  number_of_parameters: uint64  
  ? setup_parameters: [* $MOQTSetupParameter]  
}
```

Example JSON

```
{  
  "time":0.230418,"name":"moqt:control_message_created","data":{"stream_id":0,  
  "length":19,"message":{"type":"client_setup","number_of_supported_versions":1,"  
  supported_versions":[4278190088],"number_of_parameters":2,"setup_paramete  
  rs":[{"name":"max_subscribe_id","value":100},{"name":"path","value":""}]}}}
```

Implementation and interop

- quiche project [qlog Rust crate](#) supports serialization and deserialization
 - Working branch to add MOQT qlog on [GitHub](#)
- Lucas' parser tool (not open source, yet)
 - Processes logs to generate text or HTML reports, plus charts
- Mathis Engelbart's MOQ
 - Produces MOQT qlog
- Lorenzo Miniero's MOQ
 - Produces MOQT qlog, optionally combined with QUIC qlog
- Ironed out many nits in the schema, and our implementations of qlog
 - Published a new -01 on 2024-03-17
- [qvis](#) - YMMV
 - File header parsing based on older qlog spec (-08 or older).
 - ...but should be able to deal with MOQT events on top of that

Example log parser output

time	category	Type	details
0.131654	moqt	stream_type_set	owner=local, id=0, ty=control,
0.230418	moqt	control_message_created	id=0, CLIENT_SETUP {number_of_supported_versions=1, supported_versions=[4
1.450755	moqt	control_message_parsed	id=0, SERVER_SETUP {selected_version=4278190088, number_of_parameters=1, s
1.52587	moqt	control_message_parsed	id=0, ANNOUNCE {track_namespace=[value=clock,], number_of_parameters=0, p
1.619622	moqt	control_message_created	id=0, SUBSCRIBE {subscribe_id=0, track_alias=0, track_namespace=[value=cl
1.725039	moqt	control_message_created	id=0, ANNOUNCE_ERROR {track_namespace=[value=clock,], error_code=0, reaso
2.301565	moqt	control_message_parsed	id=0, SUBSCRIBE_OK {subscribe_id=0, expires=0, group_order=1, content_exi
86.66933	moqt	stream_type_set	owner=remote, id=3, ty=subgroup_header,
86.77708	moqt	subgroup_object_parsed	id=3, group_id=18, subgroup_id=0, object_id=0, extension_headers_length=0
186.50853	moqt	stream_type_set	owner=remote, id=7, ty=subgroup_header,
186.62462	moqt	subgroup_object_parsed	id=7, group_id=19, subgroup_id=0, object_id=0, extension_headers_length=0
287.07056	moqt	stream_type_set	owner=remote, id=11, ty=subgroup_header,
287.17566	moqt	subgroup_object_parsed	id=11, group_id=20, subgroup_id=0, object_id=0, extension_headers_length=0
387.29028	moqt	stream_type_set	owner=remote, id=15, ty=subgroup_header,
387.36523	moqt	subgroup_object_parsed	id=15, group_id=21, subgroup_id=0, object_id=0, extension_headers_length=0
486.5013	moqt	stream_type_set	owner=remote, id=19, ty=subgroup_header,
486.59897	moqt	subgroup_object_parsed	id=19, group_id=22, subgroup_id=0, object_id=0, extension_headers_length=0
586.62462	moqt	stream_type_set	owner=remote, id=23, ty=subgroup_header,

Example log parser output

List of events

This page lists all the events that were present in a log file.

Search: ✕

time	category	Type	details
1.450755	moqt	control_message_parsed	id=0, SERVER_SETUP {selected_version=4278190088, number_of_parameters=1, setup_parameters=[max_subscribe_id=100,]}
1.52587	moqt	control_message_parsed	id=0, ANNOUNCE {track_namespace=[value=clock,], number_of_parameters=0, parameters=[]}
2.301565	moqt	control_message_parsed	id=0, SUBSCRIBE_OK {subscribe_id=0, expires=0, group_order=1, content_exists=0, number_of_parameters=0, subscribe_parameters=[]}
10087.814	moqt	control_message_parsed	id=0, SUBSCRIBE_DONE {subscribe_id=0, status_code=4, stream_count=100, reason=unsubscribed, }

Showing 1 to 4 of 4 entries (filtered from 209 total entries)

Next steps

People have asked for MOQT qlog on and off for ~ a year

A lot of the spec groundwork is now done

Is this something we should formally adopt?