

Media Over QUIC  
Internet-Draft  
Intended status: Informational  
Expires: 8 January 2026

A. Frindell  
Meta  
7 July 2025

MoQT Test  
draft-afrind-moq-test-01

## Abstract

This document specifies the moq-test protocol, a testing protocol for Media over QUIC (MOQ) implementations. moq-test utilizes the Track Namespace as parameters to the publisher, enabling flexible and customizable testing scenarios.

## About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at <https://afrind.github.io/moq-test/draft-afrind-moq-test.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-afrind-moq-test/>.

Discussion of this document takes place on the Media Over QUIC mailing list (<mailto:moq@ietf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/moq/>. Subscribe at <https://www.ietf.org/mailman/listinfo/moq/>.

Source for this draft and an issue tracker can be found at <https://github.com/afrind/moq-test>.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 8 January 2026.

## Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Track Namespace Parameters . . . . .	3
2.1. Tuple Field 0: "moq-test-00" . . . . .	3
2.2. Tuple Field 1: Forwarding Preference . . . . .	3
2.3. Tuple Field 2: Start Group . . . . .	3
2.4. Tuple Field 3: Start Object . . . . .	4
2.5. Tuple Field 4: Last Group in Track . . . . .	4
2.6. Tuple Field 5: Last Object in Track . . . . .	4
2.7. Tuple Field 6: Objects per Group . . . . .	4
2.8. Tuple Field 7: Size of Object 0 . . . . .	4
2.9. Tuple Field 8: Size of Objects > 0 . . . . .	4
2.10. Tuple Field 9: Object Frequency . . . . .	4
2.11. Tuple Field 10: Group Increment . . . . .	4
2.12. Tuple Field 11: Object Increment . . . . .	5
2.13. Tuple Field 12: Send End of Group Markers . . . . .	5
2.14. Tuple Field 13: Test Integer Extension . . . . .	5
2.15. Tuple Field 14: Test Variable Extensions . . . . .	5
2.16. Tuple Field 15: Publisher Delivery Timeout . . . . .	5
3. Publisher Behavior . . . . .	5
4. Conventions and Definitions . . . . .	6
5. Security Considerations . . . . .	6
6. IANA Considerations . . . . .	6
7. Normative References . . . . .	6
Acknowledgments . . . . .	6
Author's Address . . . . .	7

## 1. Introduction

moq-test is designed to provide a comprehensive testing framework for MOQ implementations. By leveraging the Track Namespace, moq-test allows testers to specify various parameters that control the testing process, such as forwarding preferences, group and object ranges, and object sizes.

## 2. Track Namespace Parameters

The moq-test protocol utilizes the Track Namespace to convey testing parameters from the publisher to the subscriber. The Track Namespace tuple fields are defined as follows. Exactly 16 tuple fields must be provided, but any blank tuple field other than field 0 indicates the default value is used. Future versions will use a different value for tuple field 0 and potentially define a different set of parameters.

### 2.1. Tuple Field 0: "moq-test-00"

This field indicates that the Track Namespace parameters are for the moq-test protocol.

### 2.2. Tuple Field 1: Forwarding Preference

This field specifies the forwarding preference for the track. The following values are defined:

- \* 0 := One subgroup per group
- \* 1 := One subgroup per object
- \* 2 := Two subgroups per group (0 and 1)
- \* 3 := Datagram

The default value is 0. When using two subgroups per group, even numbered objects are sent on subgroup 0 and odd numbered objects are sent on group 1.

### 2.3. Tuple Field 2: Start Group

This field specifies the starting group number for the track. The default value is 0.

#### 2.4. Tuple Field 3: Start Object

This field specifies the starting object number for each group in the track. The default value is 0.

#### 2.5. Tuple Field 4: Last Group in Track

This field specifies the last group number in the track. The default value is  $2^{62} - 1$  (eg: the track continues until unsubscribed).

Use caution when setting this field accessing this track via FETCH.

#### 2.6. Tuple Field 5: Last Object in Track

This field specifies the last object number in the track. The maximum value is the number of objects in the group, plus one if end of group markers are used.

The default value is the maximum value.

#### 2.7. Tuple Field 6: Objects per Group

This field specifies the number of objects per group. The default is 10.

#### 2.8. Tuple Field 7: Size of Object 0

This field specifies the size of the first object in bytes. The default is 1024 bytes.

#### 2.9. Tuple Field 8: Size of Objects > 0

This specifies the size of objects with ID greater than 0, in bytes. The default value is 100 bytes.

#### 2.10. Tuple Field 9: Object Frequency

This field specifies the frequency at which objects are sent in milliseconds. The default value is 1000.

#### 2.11. Tuple Field 10: Group Increment

This field specifies the increment between group numbers. The default is 1.

### 2.12. Tuple Field 11: Object Increment

This tuple field specifies the increment between object numbers. The default is 1.

### 2.13. Tuple Field 12: Send End of Group Markers

This field specifies whether to send end of group markers.

\* 0 := Do not send End of Group markers

\* 1 := Send End of Group markers

End of group markers are an Object with status 0x3 End of Group.

The default value is 0 / No End of Group markers.

### 2.14. Tuple Field 13: Test Integer Extension

This field specifies whether to include a test integer extension on every object in the track. The ID of the extension is 2 \* field value. The value of the extension is random.

The default value indicates no integer extension is sent.

### 2.15. Tuple Field 14: Test Variable Extensions

This field specifies whether to include a test variable extension on every object in the track. The ID of the extension is 2 \* field value + 1. The value of the extension is random data.

The default value indicates no variable extension is sent.

### 2.16. Tuple Field 15: Publisher Delivery Timeout

This field specifies the publisher delivery timeout in milliseconds. The default is no delivery timeout.

## 3. Publisher Behavior

If any namespace tuple field other than zero is not a positive integer, or is outside the supported range of the publisher, a SUBSCRIBE\_ERROR or FETCH\_ERROR is returned.

Upon receiving a valid SUBSCRIBE or FETCH for a moq-test track, the publisher MUST publish objects according to the parameters of the namespace.

Upon receiving a SUBSCRIBE\_ANNOUNCES with a Track Namespace, the publisher MUST use PUBLISH to send objects according to the parameters of the namespace. If the publisher receives a PUBLISH\_ERROR, it will stop sending objects immediately.

The payload of each published object is the character t (for test) repeated for the length of the payload.

When Datagram forwarding preference is used, certain combinations of options result in objects that cannot be transmitted below the maximum datagram size. The publisher either responds with SUBSCRIBE\_ERROR if detected when handling the SUBSCRIBE, or SUBSCRIBE\_DONE with an error code if detected when publishing the object.

#### 4. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

#### 5. Security Considerations

The subscriber can generate an asymmetric amount of work on the server. Servers SHOULD set limits on certain values such as object frequency and object size to prevent resource exhaustion attacks.

#### 6. IANA Considerations

This document has no IANA actions.

#### 7. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.

#### Acknowledgments

Members of the Media over QUIC working group

Author's Address

Alan Frindell  
Meta  
Email: [afrind@meta.com](mailto:afrind@meta.com)