

# KMIP Additional Message Encodings Version 1.0

## Committee Specification Draft 02

19 June 2014

### Specification URIs

#### This version:

<http://docs.oasis-open.org/kmip/kmip-addtl-msg-enc/v1.0/csd02/kmip-addtl-msg-enc-v1.0-csd02.doc> (Authoritative)  
<http://docs.oasis-open.org/kmip/kmip-addtl-msg-enc/v1.0/csd02/kmip-addtl-msg-enc-v1.0-csd02.html>  
<http://docs.oasis-open.org/kmip/kmip-addtl-msg-enc/v1.0/csd02/kmip-addtl-msg-enc-v1.0-csd02.pdf>

#### Previous version:

<http://docs.oasis-open.org/kmip/kmip-addtl-msg-enc/v1.0/csprd01/kmip-addtl-msg-enc-v1.0-csprd01.doc> (Authoritative)  
<http://docs.oasis-open.org/kmip/kmip-addtl-msg-enc/v1.0/csprd01/kmip-addtl-msg-enc-v1.0-csprd01.html>  
<http://docs.oasis-open.org/kmip/kmip-addtl-msg-enc/v1.0/csprd01/kmip-addtl-msg-enc-v1.0-csprd01.pdf>

#### Latest version:

<http://docs.oasis-open.org/kmip/kmip-addtl-msg-enc/v1.0/kmip-addtl-msg-enc-v1.0.doc> (Authoritative)  
<http://docs.oasis-open.org/kmip/kmip-addtl-msg-enc/v1.0/kmip-addtl-msg-enc-v1.0.html>  
<http://docs.oasis-open.org/kmip/kmip-addtl-msg-enc/v1.0/kmip-addtl-msg-enc-v1.0.pdf>

### Technical Committee:

OASIS Key Management Interoperability Protocol (KMIP) TC

### Chairs:

Subhash Sankuratripati ([Subhash.Sankuratripati@netapp.com](mailto:Subhash.Sankuratripati@netapp.com)), NetApp  
Saikat Saha ([saikat.saha@oracle.com](mailto:saikat.saha@oracle.com)), Oracle

### Editor:

Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com)), Cryptsoft Pty Ltd.

### Related work:

This specification is related to:

- *Key Management Interoperability Protocol Profiles Version 1.2*. Edited by Tim Hudson and Robert Lockhart. Latest version: <http://docs.oasis-open.org/kmip/profiles/v1.2/kmip-profiles-v1.2.html>.
- *Key Management Interoperability Protocol Specification Version 1.2*. Edited by Kiran Thota and Kelley Burgin. Latest version: <http://docs.oasis-open.org/kmip/spec/v1.2/kmip-spec-v1.2.html>.
- *Key Management Interoperability Protocol Test Cases Version 1.2*. Edited by Tim Hudson and Faisal Faruqui. Latest version: <http://docs.oasis-open.org/kmip/testcases/v1.2/kmip-testcases-v1.2.html>.

- *Key Management Interoperability Protocol Usage Guide Version 1.2*. Edited by Indra Fitzgerald and Judith Furlong. Latest version: <http://docs.oasis-open.org/kmip/ug/v1.2/kmip-ug-v1.2.html>.

**Abstract:**

Describes additional (optional) message encodings as an alternative to the (mandatory) raw TTLV (Tag, Type, Length, Value) encoding including HTTPS, JSON and XML.

**Status:**

This document was last revised or approved by the OASIS Key Management Interoperability Protocol (KMIP) TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at <https://www.oasis-open.org/committees/kmip/>.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (<https://www.oasis-open.org/committees/kmip/ipr.php>).

**Citation format:**

When referencing this specification the following citation format should be used:

**[kmip-addtl-msg-enc-v1.0]**

*KMIP Additional Message Encodings Version 1.0*. Edited by Tim Hudson. 19 June 2014. OASIS Committee Specification Draft 02. <http://docs.oasis-open.org/kmip/kmip-addtl-msg-enc/v1.0/csd02/kmip-addtl-msg-enc-v1.0-csd02.html>. Latest version: <http://docs.oasis-open.org/kmip/kmip-addtl-msg-enc/v1.0/kmip-addtl-msg-enc-v1.0.html>.

---

# Notices

Copyright © OASIS Open 2014. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full [Policy](#) may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of [OASIS](#), the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <https://www.oasis-open.org/policies-guidelines/trademark> for above guidance.

---

# Table of Contents

1	Introduction .....	6
1.1	Terminology .....	6
1.2	Normative References .....	6
1.3	Non-Normative References .....	7
2	HTTPS Profile.....	8
2.1	Authentication Suite.....	8
2.2	KMIP Port Number.....	8
2.3	Request URI .....	8
2.4	HTTP Encoding - Client.....	8
2.5	HTTP Encoding - Server.....	8
3	HTTPS Profile Test Cases .....	10
3.1	Mandatory HTTPS Profile Test Cases KMIP v1.0.....	10
3.1.1	MSGENC-HTTPS-M-1-10 - Query, Maximum Response Size .....	10
3.2	Mandatory HTTPS Profile Test Cases KMIP v1.1 .....	14
3.2.1	MSGENC-HTTPS-M-1-11 - Query, Maximum Response Size .....	14
3.3	Mandatory HTTPS Profile Test Cases KMIP v1.2.....	18
3.3.1	MSGENC-HTTPS-M-1-12 - Query, Maximum Response Size .....	18
4	JSON Profile.....	23
4.1	JSON Encoding .....	23
4.1.1	Hex representations .....	23
4.1.2	Tags.....	23
4.1.3	Normalizing Names .....	23
4.1.4	Type.....	24
4.1.5	Value .....	24
4.1.6	JSON Object.....	25
5	JSON Profile Test Cases .....	27
5.1	Mandatory JSON Profile Test Cases KMIP v1.0 .....	27
5.1.1	MSGENC-JSON-M-1-10 - Query, Maximum Response Size .....	27
5.2	Mandatory JSON Profile Test Cases KMIP v1.1 .....	31
5.2.1	MSGENC-JSON-M-1-11 - Query, Maximum Response Size .....	31
5.3	Mandatory JSON Profile Test Cases KMIP v1.2.....	35
5.3.1	MSGENC-JSON-M-1-12 - Query, Maximum Response Size .....	35
6	XML Profile .....	40
6.1	XML Encoding .....	40
6.1.1	Hex representations .....	40
6.1.2	Tags.....	40
6.1.3	Normalizing Names .....	40
6.1.4	Type.....	41
6.1.5	Value .....	41
6.1.6	XML Element Encoding.....	42
7	XML Profile Test Cases.....	44
7.1	Mandatory XML Profile Test Cases KMIP v1.0 .....	44
7.1.1	MSGENC-XML-M-1-10 - Query, Maximum Response Size .....	44

7.2 Mandatory XML Profile Test Cases KMIP v1.1 .....	46
7.2.1 MSGENC-XML-M-1-11 - Query, Maximum Response Size .....	46
7.3 Mandatory XML Profile Test Cases KMIP v1.2 .....	49
7.3.1 MSGENC-XML-M-1-12 - Query, Maximum Response Size .....	49
8 Conformance .....	53
8.1 HTTPS Profile .....	53
8.1.1 HTTPS Client KMIP v1.0 Profile Conformance .....	53
8.1.2 HTTPS Client KMIP v1.1 Profile Conformance .....	53
8.1.3 HTTPS Client KMIP v1.2 Profile Conformance .....	53
8.1.4 HTTPS Server KMIP v1.0 Profile Conformance .....	53
8.1.5 HTTPS Server KMIP v1.1 Profile Conformance .....	53
8.1.6 HTTPS Server KMIP v1.2 Profile Conformance .....	54
8.2 JSON Profile .....	54
8.2.1 JSON Client KMIP v1.0 Profile Conformance .....	54
8.2.2 JSON Client KMIP v1.1 Profile Conformance .....	54
8.2.3 JSON Client KMIP v1.2 Profile Conformance .....	54
8.2.4 JSON Server KMIP v1.0 Profile Conformance .....	54
8.2.5 JSON Server KMIP v1.1 Profile Conformance .....	55
8.2.6 JSON Server KMIP v1.2 Profile Conformance .....	55
8.3 XML Profile .....	55
8.3.1 XML Client KMIP v1.0 Profile Conformance .....	55
8.3.2 XML Client KMIP v1.1 Profile Conformance .....	55
8.3.3 XML Client KMIP v1.2 Profile Conformance .....	56
8.3.4 XML Server KMIP v1.0 Profile Conformance .....	56
8.3.5 XML Server KMIP v1.1 Profile Conformance .....	56
8.3.6 XML Server KMIP v1.2 Profile Conformance .....	56
8.4 Permitted Test Case Variations .....	56
8.4.1 Variable Items .....	57
8.4.2 Variable behavior .....	58
Appendix A. Acknowledgments .....	59
Appendix B. KMIP Specification Cross Reference .....	62
Appendix C. Revision History .....	67

# 1 Introduction

For normative definition of the elements of KMIP see the [KMIP Specification](#) [KMIP-SPEC] and the [KMIP Profiles](#) [KMIP-PROF].

This profile defines the necessary encoding rules for the transport of KMIP TTLV messages encoded in:

- [Hypertext Transfer Protocol](#) [RFC2616] over [TLS](#) as specified in [HTTP over TLS](#) [RFC2818]
- [JavaScript Object Notification](#) [RFC4627]
- [Extensible Markup Language](#) [XML]

## 1.1 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in **Error! Reference source not found.**

## 1.2 Normative References

- [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- [RFC2616] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, *Hypertext Transfer Protocol -- HTTP/1.1*, <http://www.ietf.org/rfc/rfc2616.txt>, IETF RFC 2616, June 1999.
- [RFC2818] E. Rescorla, *HTTP over TLS*, IETF RFC 2818, May 2000, <http://www.ietf.org/rfc/rfc2818.txt>
- [RFC7159] Bray, T., Ed., *The JavaScript Object Notation (JSON) Data Interchange Format*, RFC 7159, March 2014. <http://www.ietf.org/rfc/rfc7159.txt>
- [XML] Bray, Tim, et.al. eds, *Extensible Markup Language (XML) 1.0 (Fifth Edition)*, W3C Recommendation 26 November 2008, available at <http://www.w3.org/TR/2008/REC-xml-20081126/>
- [KMIP-SPEC] One or more of [KMIP-SPEC-1\_0], [KMIP-SPEC-1\_1], [KMIP-SPEC-1\_2]
- [KMIP-SPEC-1\_0] Key Management Interoperability Protocol Specification Version 1.0 <http://docs.oasis-open.org/kmip/spec/v1.0/os/kmip-spec-1.0-os.doc> OASIS Standard, October 2010.
- [KMIP-SPEC-1\_1] *Key Management Interoperability Protocol Specification Version 1.1*. <http://docs.oasis-open.org/kmip/spec/v1.1/os/kmip-spec-v1.1-os.doc> OASIS Standard. 24 January 2013.
- [KMIP-SPEC-1\_2] *Key Management Interoperability Protocol Specification Version 1.2*. [URL](#) Candidate OASIS Standard 01. **DD MMM YYYY**.
- [KMIP-PROF] One or more of [KMIP-PROF-1\_0], [KMIP-PROF-1\_1], [KMIP-PROF-1\_2]
- [KMIP-PROF-1\_0] *Key Management Interoperability Protocol Profiles Version 1.0*. <http://docs.oasis-open.org/kmip/profiles/v1.0/os/kmip-profiles-1.0-os.doc> OASIS Standard. 1 October 2010.
- [KMIP-PROF-1\_1] *Key Management Interoperability Protocol Profiles Version 1.1*. <http://docs.oasis-open.org/kmip/profiles/v1.1/os/kmip-profiles-v1.1-os.doc> OASIS Standard 01. 24 January 2013.
- [KMIP-PROF-1\_2] *Key Management Interoperability Protocol Profiles Version 1.2*. [URL](#) Candidate OASIS Standard 01. **DD MMM YYYY**.

## 1.3 Non-Normative References

- [XML-SCHEMA]** Paul V. Biron, Ashok Malhotra, XML Schema Part 2: Datatypes Second Edition, W3C Recommendation 26 November 2008, available at <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>

---

## 2 HTTPS Profile

The Hypertext Transfer Protocol over Transport Layer Security (HTTPS) is simply the use of HTTP over TLS in the same manner that HTTP is used over TCP.

KMIP over HTTPS is simply the use of KMIP messages over HTTPS in the same manner that KMIP is used over TLS.

### 2.1 Authentication Suite

Implementations conformant to this profile SHALL support one or more of the Authentication Suites defined within section 3 of [KMIP-PROF].

### 2.2 KMIP Port Number

KMIP servers conformant to this profile MAY use TCP port number 5696, as assigned by IANA, to receive and send KMIP messages provided that both HTTP and non-HTTP encoded messages are supported.

KMIP clients SHALL enable end user configuration of the TCP port number used, as a KMIP server may specify a different TCP port number.

### 2.3 Request URI

KMIP servers conformant to this profile SHOULD support the value */kmip* as the target URI.

KMIP clients SHALL enable end user configuration of the target URI used as a KMIP server may specify a different target URI.

### 2.4 HTTP Encoding - Client

KMIP client implementations conformant to this profile:

1. SHALL support HTTP/1.0 and/or HTTP/1.1 over TLS conformant to [RFC2818]
2. SHALL use the POST request method
3. SHALL specify a Content-Type of "application/octet-stream" if the message encoding is TTLV
4. SHALL specify a Content-Type of "text/xml" if the message encoding is XML
5. SHALL specify a Content-Type of "application/json" if the message encoding is JSON
6. SHALL specify a Content-Length
7. SHALL specify a Cache-Control of "no-cache"
8. SHALL send KMIP TTLV message in binary format as the body of the HTTP request

KMIP clients that support responding to server to client operations SHALL behave as a HTTPS server.

### 2.5 HTTP Encoding - Server

KMIP server implementations conformant to this profile:

1. SHALL support HTTP/1.0 and HTTP/1.1 over TLS conformant to [RFC2818]
2. SHALL return HTTP response code 200 if a KMIP response is available
3. SHALL specify a Content-Type of "application/octet-stream" if the message encoding is TTLV
4. SHALL specify a Content-Type of "text/xml" if the message encoding is XML
5. SHALL specify a Content-Type of "application/json" if the message encoding is JSON
6. SHALL specify a Content-Length



- 88        7. SHALL specify a Cache-Control of “no-cache”  
89        8. SHALL send KMIP TTLV message in binary format as the body of the HTTP request  
90  
91        KMIP servers that support server to client operations SHALL behave as a HTTPS client.  
92

### 3 HTTPS Profile Test Cases

The test cases define a number of request-response pairs for KMIP operations. Each test case is provided in the XML format specified in section 6 intended to be both human-readable and usable by automated tools. The time sequence (starting from 0) for each request-response pair is noted and line numbers are provided for ease of cross-reference for a given test sequence.

Each test case has a unique label (the section name) which includes indication of mandatory (-M-) or optional (-O-) status and the protocol version major and minor numbers as part of the identifier.

The test cases may depend on a specific configuration of a KMIP client and server being configured in a manner consistent with the test case assumptions.

Where possible the flow of unique identifiers between tests, the date-time values, and other dynamic items are indicated using symbolic identifiers – in actual request and response messages these dynamic values will be filled in with valid values.

Note: the values for the returned items and the custom attributes are illustrative. Actual values from a real client system may vary as specified in section 8.4

#### 3.1 Mandatory HTTPS Profile Test Cases KMIP v1.0

##### 3.1.1 MSGENC-HTTPS-M-1-10 - Query, Maximum Response Size

Perform a Query operation, querying the Operations and Objects supported by the server, with a restriction on the Maximum Response Size set in the request header. Since the resulting Query response is too big, an error is returned. Increase the Maximum Response Size, resubmit the Query request, and get a successful response.

The specific list of operations and object types returned in the response MAY vary.

	# TIME 0
0001	<RequestMessage>
0002	<RequestHeader>
0003	<ProtocolVersion>
0004	<ProtocolVersionMajor type="Integer" value="1"/>
0005	<ProtocolVersionMinor type="Integer" value="0"/>
0006	</ProtocolVersion>
0007	<MaximumResponseSize type="Integer" value="256"/>
0008	<BatchCount type="Integer" value="1"/>
0009	</RequestHeader>
0010	<BatchItem>
0011	<Operation type="Enumeration" value="Query"/>
0012	<RequestPayload>
0013	<QueryFunction type="Enumeration" value="QueryOperations"/>
0014	<QueryFunction type="Enumeration" value="QueryObjects"/>
0015	</RequestPayload>
0016	</BatchItem>
0017	</RequestMessage>
42007801000000904200770100000048420069010000002042006a02000000040000000100000000	
42006b020000000400000000000000004200500200000004000001000000000042000d0200000004	
000000010000000042000f010000003842005c050000000400000018000000004200790100000020	
4200740500000004000000010000000042007405000000040000000200000000	
00000000:	50 4f 53 54 20 2f 6b 6d-69 70 20 48 54 54 50 2f POST /kmip HTTP/
00000010:	31 2e 30 0d 0a 50 72 61-67 6d 61 3a 20 6e 6f 2d 1.0..Pragma: no-
00000020:	63 61 63 68 65 0d 0a 43-61 63 68 65 2d 43 6f 6e cache..Cache-Con
00000030:	74 72 6f 6c 3a 20 6e 6f-2d 63 61 63 68 65 0d 0a trol: no-cache..
00000040:	43 6f 6e 6e 65 63 74 69-6f 6e 3a 20 6b 65 65 70 Connection: keep
00000050:	2d 61 6c 69 76 65 0d 0a-43 6f 6e 74 65 6e 74 2d -alive..Content-

	<pre> 00000060: 54 79 70 65 3a 20 61 70-70 6c 69 63 61 74 69 6f  Type: applicatio 00000070: 6e 2f 6f 63 74 65 74 2d-73 74 72 65 61 6d 0d 0a  n/octet-stream.. 00000080: 43 6f 6e 74 65 6e 74 2d-4c 65 6e 67 74 68 3a 20  Content-Length: 00000090: 31 35 32 20 20 20 20 20-20 20 0d 0a 0d 0a 42 00  152      ....B. 000000a0: 15 32 78 01 00 00 00 90-42 00 77 01 00 00 00 48  .2x.....B.w....H 000000b0: 42 00 69 01 00 00 00 20-42 00 6a 02 00 00 00 04  B.i.... B.j..... 000000c0: 00 00 00 01 00 00 00 00-42 00 6b 02 00 00 00 04  ....B.k..... 000000d0: 00 00 00 00 00 00 00 00-42 00 50 02 00 00 00 04  ....B.P..... 000000e0: 00 00 01 00 00 00 00 00-42 00 0d 02 00 00 00 04  ....B..... 000000f0: 00 00 00 01 00 00 00 00-42 00 0f 01 00 00 00 38  ....B.....8 00000100: 42 00 5c 05 00 00 00 04-00 00 00 18 00 00 00 00  B.\..... 00000110: 42 00 79 01 00 00 00 20-42 00 74 05 00 00 00 04  B.y.... B.t..... 00000120: 00 00 00 01 00 00 00 00-42 00 74 05 00 00 00 04  ....B.t..... 00000130: 00 00 00 02 00 00 00 00-      .... </pre>
0018 0019 0020 0021 0022 0023 0024 0025 0026 0027 0028 0029 0030 0031 0032	<pre> &lt;ResponseMessage&gt;   &lt;ResponseHeader&gt;     &lt;ProtocolVersion&gt;       &lt;ProtocolVersionMajor type="Integer" value="1"/&gt;       &lt;ProtocolVersionMinor type="Integer" value="0"/&gt;     &lt;/ProtocolVersion&gt;     &lt;TimeStamp type="DateTime" value="2013-06-26T09:09:17+00:00"/&gt;     &lt;BatchCount type="Integer" value="1"/&gt;   &lt;/ResponseHeader&gt; &lt;BatchItem&gt;     &lt;Operation type="Enumeration" value="Query"/&gt;     &lt;ResultStatus type="Enumeration" value="OperationFailed"/&gt;     &lt;ResultReason type="Enumeration" value="ResponseTooLarge"/&gt;     &lt;ResultMessage type="TextString" value="TOO_LARGE"/&gt;   &lt;/BatchItem&gt; &lt;/ResponseMessage&gt; </pre> <p>42007b01000000a042007a0100000048420069010000002042006a02000000040000000100000000  42006b020000000400000000000000042009209000000080000000051caafbd42000d0200000004  000000010000000042000f010000004842005c0500000004000000180000000042007f0500000004  000000010000000042007e0500000004000000020000000042007d0700000009544f4f5f4c415247  4500000000000000</p> <pre> 00000000: 48 54 54 50 2f 31 2e 31-20 32 30 30 20 4f 4b 0d  HTTP/1.1 200 OK. 00000010: 0a 43 6f 6e 74 65 6e 74-2d 54 79 70 65 3a 20 61  .Content-Type: a 00000020: 70 70 6c 69 63 61 74 69-6f 6e 2f 6f 63 74 65 74  pplication/octet 00000030: 2d 73 74 72 65 61 6d 0d-0a 43 6f 6e 74 65 6e 74  -stream.Content 00000040: 2d 4c 65 6e 67 74 68 3a-20 31 36 38 0d 0a 0d 0a  -Length: 168.... 00000050: 42 00 7b 01 00 00 00 a0-42 00 7a 01 00 00 00 48  B.{.... B.z....H 00000060: 42 00 69 01 00 00 00 20-42 00 6a 02 00 00 00 04  B.i.... B.j..... 00000070: 00 00 00 01 00 00 00 00-42 00 6b 02 00 00 00 04  ....B.k..... 00000080: 00 00 00 00 00 00 00 00-42 00 92 09 00 00 00 08  ....B..... 00000090: 00 00 00 00 51 ca af bd-42 00 0d 02 00 00 00 04  ....QJ/=B..... 000000a0: 00 00 00 01 00 00 00 00-42 00 0f 01 00 00 00 48  ....B.....H 000000b0: 42 00 5c 05 00 00 00 04-00 00 00 18 00 00 00 00  B.\..... 000000c0: 42 00 7f 05 00 00 00 04-00 00 00 01 00 00 00 00  B..... 000000d0: 42 00 7e 05 00 00 00 04-00 00 00 02 00 00 00 00  B.~..... 000000e0: 42 00 7d 07 00 00 00 09-54 4f 4f 5f 4c 41 52 47  B.}....TOO_LARG 000000f0: 45 00 00 00 00 00 00 00-      E..... </pre>
0032 0033 0034 0035 0036 0037 0038 0039 0040 0041	<pre> # TIME 1 &lt;RequestMessage&gt;   &lt;RequestHeader&gt;     &lt;ProtocolVersion&gt;       &lt;ProtocolVersionMajor type="Integer" value="1"/&gt;       &lt;ProtocolVersionMinor type="Integer" value="0"/&gt;     &lt;/ProtocolVersion&gt;     &lt;MaximumResponseSize type="Integer" value="2048"/&gt;     &lt;BatchCount type="Integer" value="1"/&gt;   &lt;/RequestHeader&gt;   &lt;BatchItem&gt; </pre>

0042	<Operation type="Enumeration" value="Query"/>
0043	<RequestPayload>
0044	<QueryFunction type="Enumeration" value="QueryOperations"/>
0045	<QueryFunction type="Enumeration" value="QueryObjects"/>
0046	</RequestPayload>
0047	</BatchItem>
0048	</RequestMessage>
	<p>42007801000000904200770100000048420069010000002042006a02000000040000000100000000</p> <p>42006b0200000004000000000000000420050020000000400000800000000042000d0200000004</p> <p>000000010000000042000f010000003842005c050000000400000018000000004200790100000020</p> <p>4200740500000004000000010000000042007405000000040000000200000000</p>
	<pre> 00000000: 50 4f 53 54 20 2f 6b 6d-69 70 20 48 54 54 50 2f  POST /kmip HTTP/ 00000010: 31 2e 30 0d 0a 50 72 61-67 6d 61 3a 20 6e 6f 2d  1.0..Pragma: no- 00000020: 63 61 63 68 65 0d 0a 43-61 63 68 65 2d 43 6f 6e  cache..Cache-Con 00000030: 74 72 6f 6c 3a 20 6e 6f-2d 63 61 63 68 65 0d 0a  trol: no-cache.. 00000040: 43 6f 6e 6e 65 63 74 69-6f 6e 3a 20 6b 65 65 70  Connection: keep 00000050: 2d 61 6c 69 76 65 0d 0a-43 6f 6e 74 65 6e 74 2d  -alive..Content- 00000060: 54 79 70 65 3a 20 61 70-70 6c 69 63 61 74 69 6f  Type: applicatio 00000070: 6e 2f 6f 63 74 65 74 2d-73 74 72 65 61 6d 0d 0a  n/octet-stream.. 00000080: 43 6f 6e 74 65 6e 74 2d-4c 65 6e 67 74 68 3a 20  Content-Length: 00000090: 31 35 32 20 20 20 20 20-20 20 0d 0a 0d 0a 42 00  152      ....B. 000000a0: 15 32 78 01 00 00 00 90-42 00 77 01 00 00 00 48  .2x.....B.w....H 000000b0: 42 00 69 01 00 00 00 20-42 00 6a 02 00 00 00 04  B.i.... B.j..... 000000c0: 00 00 00 01 00 00 00 00-42 00 6b 02 00 00 00 04  .....B.k..... 000000d0: 00 00 00 00 00 00 00 00-42 00 50 02 00 00 00 04  .....B.P..... 000000e0: 00 00 08 00 00 00 00 00-42 00 0d 02 00 00 00 04  .....B..... 000000f0: 00 00 00 01 00 00 00 00-42 00 0f 01 00 00 00 38  .....B.....8 00000100: 42 00 5c 05 00 00 00 04-00 00 18 00 00 00 00  B.\..... 00000110: 42 00 79 01 00 00 00 20-42 00 74 05 00 00 00 04  B.y.... B.t..... 00000120: 00 00 00 01 00 00 00 00-42 00 74 05 00 00 00 04  .....B.t..... 00000130: 00 00 00 02 00 00 00 00-          ..... </pre>
0049	<ResponseMessage>
0050	<ResponseHeader>
0051	<ProtocolVersion>
0052	<ProtocolVersionMajor type="Integer" value="1"/>
0053	<ProtocolVersionMinor type="Integer" value="0"/>
0054	</ProtocolVersion>
0055	<TimeStamp type="DateTime" value="2013-06-26T09:09:17+00:00"/>
0056	<BatchCount type="Integer" value="1"/>
0057	</ResponseHeader>
0058	<BatchItem>
0059	<Operation type="Enumeration" value="Query"/>
0060	<ResultStatus type="Enumeration" value="Success"/>
0061	<ResponsePayload>
0062	<Operation type="Enumeration" value="Query"/>
0063	<Operation type="Enumeration" value="Locate"/>
0064	<Operation type="Enumeration" value="Destroy"/>
0065	<Operation type="Enumeration" value="Get"/>
0066	<Operation type="Enumeration" value="Create"/>
0067	<Operation type="Enumeration" value="Register"/>
0068	<Operation type="Enumeration" value="GetAttributes"/>
0069	<Operation type="Enumeration" value="GetAttributeList"/>
0070	<Operation type="Enumeration" value="AddAttribute"/>
0071	<Operation type="Enumeration" value="ModifyAttribute"/>
0072	<Operation type="Enumeration" value="DeleteAttribute"/>
0073	<Operation type="Enumeration" value="Activate"/>
0074	<Operation type="Enumeration" value="Revoke"/>
0075	<Operation type="Enumeration" value="Poll"/>
0076	<Operation type="Enumeration" value="Cancel"/>
0077	<Operation type="Enumeration" value="Check"/>

0078	<Operation type="Enumeration" value="GetUsageAllocation"/>
0079	<Operation type="Enumeration" value="CreateKeyPair"/>
0080	<Operation type="Enumeration" value="ReKey"/>
0081	<Operation type="Enumeration" value="Archive"/>
0082	<Operation type="Enumeration" value="Recover"/>
0083	<Operation type="Enumeration" value="ObtainLease"/>
0084	<Operation type="Enumeration" value="Certify"/>
0085	<Operation type="Enumeration" value="ReCertify"/>
0086	<Operation type="Enumeration" value="Notify"/>
0087	<Operation type="Enumeration" value="Put"/>
0088	<ObjectType type="Enumeration" value="Certificate"/>
0089	<ObjectType type="Enumeration" value="SymmetricKey"/>
0090	<ObjectType type="Enumeration" value="SecretData"/>
0091	<ObjectType type="Enumeration" value="PublicKey"/>
0092	<ObjectType type="Enumeration" value="PrivateKey"/>
0093	<ObjectType type="Enumeration" value="Template"/>
0094	<ObjectType type="Enumeration" value="OpaqueObject"/>
0095	<ObjectType type="Enumeration" value="SplitKey"/>
0096	</ResponsePayload>
0097	</BatchItem>
0098	</ResponseMessage>
42007b01000002a042007a0100000048420069010000002042006a02000000040000000100000000 42006b02000000040000000000000004200920900000008000000051caafbd42000d0200000004 000000010000000042000f010000024842005c0500000004000000180000000042007f0500000004 000000000000000042007c010000022042005c0500000004000000180000000042005c0500000004 000000080000000042005c0500000004000000140000000042005c05000000040000000a00000000 42005c0500000004000000010000000042005c0500000004000000030000000042005c0500000004 0000000b0000000042005c05000000040000000c0000000042005c05000000040000000d00000000 42005c05000000040000000e0000000042005c05000000040000000f0000000042005c0500000004 000000120000000042005c0500000004000000130000000042005c05000000040000001a00000000 42005c0500000004000000190000000042005c0500000004000000090000000042005c0500000004 000000110000000042005c0500000004000000020000000042005c05000000040000000700000000 42005c0500000004000000150000000042005c0500000004000000160000000042005c0500000004 000000100000000042005c0500000004000000060000000042005c05000000040000000700000000 42005c05000000040000001b0000000042005c05000000040000001c000000004200570500000004 00000010000000004200570500000004000000020000000042005705000000040000000700000000 42005705000000040000000300000000420057050000000400000004000000004200570500000004 00000006000000004200570500000004000000080000000042005705000000040000000500000000	
00000000: 48 54 54 50 2f 31 2e 31-20 32 30 30 20 4f 4b 0d HTTP/1.1 200 OK. 00000010: 0a 43 6f 6e 74 65 6e 74-2d 54 79 70 65 3a 20 61 .Content-Type: a 00000020: 70 70 6c 69 63 61 74 69-6f 6e 2f 6f 63 74 65 74 pplication/octet 00000030: 2d 73 74 72 65 61 6d 0d-0a 43 6f 6e 74 65 6e 74 -stream..Content 00000040: 2d 4c 65 6e 67 74 68 3a-20 36 38 30 0d 0a 0d 0a -Length: 680.... 00000050: 42 00 7b 01 00 00 02 a0-42 00 7a 01 00 00 00 48 B.{.... B.z....H 00000060: 42 00 69 01 00 00 00 20-42 00 6a 02 00 00 00 04 B.i.... B.j..... 00000070: 00 00 00 01 00 00 00 00-42 00 6b 02 00 00 00 04 .....B.k..... 00000080: 00 00 00 00 00 00 00 00-42 00 92 09 00 00 00 08 .....B..... 00000090: 00 00 00 00 51 ca af bd-42 00 0d 02 00 00 00 04 ....QJ/=B..... 000000a0: 00 00 00 01 00 00 00 00-42 00 0f 01 00 00 02 48 .....B.....H 000000b0: 42 00 5c 05 00 00 00 04-00 00 18 00 00 00 00 B.\..... 000000c0: 42 00 7f 05 00 00 00 04-00 00 00 00 00 00 00 B..... 000000d0: 42 00 7c 01 00 00 02 20-42 00 5c 05 00 00 00 04 B. .... B.\..... 000000e0: 00 00 00 18 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 000000f0: 00 00 00 08 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 00000100: 00 00 00 14 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 00000110: 00 00 00 0a 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 00000120: 00 00 00 01 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 00000130: 00 00 00 03 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 00000140: 00 00 00 0b 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 00000150: 00 00 00 0c 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 00000160: 00 00 00 0d 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 00000170: 00 00 00 0e 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 00000180: 00 00 00 0f 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 00000190: 00 00 00 12 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 000001a0: 00 00 00 13 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\.....	

000001b0:	00 00 00 1a 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000001c0:	00 00 00 19 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000001d0:	00 00 00 09 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000001e0:	00 00 00 11 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000001f0:	00 00 00 02 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000200:	00 00 00 04 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000210:	00 00 00 15 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000220:	00 00 00 16 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000230:	00 00 00 10 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000240:	00 00 00 06 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000250:	00 00 00 07 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000260:	00 00 00 1b 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000270:	00 00 00 1c 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
00000280:	00 00 00 01 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
00000290:	00 00 00 02 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
000002a0:	00 00 00 07 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
000002b0:	00 00 00 03 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
000002c0:	00 00 00 04 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
000002d0:	00 00 00 06 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
000002e0:	00 00 00 08 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
000002f0:	00 00 00 05 00 00 00 00-	.....

114

## 115 3.2 Mandatory HTTPS Profile Test Cases KMIP v1.1

### 116 3.2.1 MSGENC-HTTPS-M-1-11 - Query, Maximum Response Size

117 Perform a Query operation, querying the Operations and Objects supported by the server, with a  
118 restriction on the Maximum Response Size set in the request header. Since the resulting Query response  
119 is too big, an error is returned. Increase the Maximum Response Size, resubmit the Query request, and  
120 get a successful response.

121 The specific list of operations and object types returned in the response MAY vary.

	# TIME 0
0001	<RequestMessage>
0002	<RequestHeader>
0003	<ProtocolVersion>
0004	<ProtocolVersionMajor type="Integer" value="1"/>
0005	<ProtocolVersionMinor type="Integer" value="1"/>
0006	</ProtocolVersion>
0007	<MaximumResponseSize type="Integer" value="256"/>
0008	<BatchCount type="Integer" value="1"/>
0009	</RequestHeader>
0010	<BatchItem>
0011	<Operation type="Enumeration" value="Query"/>
0012	<RequestPayload>
0013	<QueryFunction type="Enumeration" value="QueryOperations"/>
0014	<QueryFunction type="Enumeration" value="QueryObjects"/>
0015	</RequestPayload>
0016	</BatchItem>
0017	</RequestMessage>
	42007801000000904200770100000048420069010000002042006a02000000040000000100000000
	42006b02000000040000000010000000042005002000000040000001000000000042000d0200000004
	0000000100000000042000f010000003842005c050000000400000018000000004200790100000020
	4200740500000004000000010000000042007405000000040000000200000000
	00000000: 50 4f 53 54 20 2f 6b 6d-69 70 20 48 54 54 50 2f POST /kmip HTTP/
	00000010: 31 2e 30 0d 0a 50 72 61-67 6d 61 3a 20 6e 6f 2d 1.0..Pragma: no-
	00000020: 63 61 63 68 65 0d 0a 43-61 63 68 65 2d 43 6f 6e cache..Cache-Con
	00000030: 74 72 6f 6c 3a 20 6e 6f-2d 63 61 63 68 65 0d 0a trol: no-cache..
	00000040: 43 6f 6e 6e 65 63 74 69-6f 6e 3a 20 6b 65 65 70 Connection: keep
	00000050: 2d 61 6c 69 76 65 0d 0a-43 6f 6e 74 65 6e 74 2d -alive..Content-

	<pre> 00000060: 54 79 70 65 3a 20 61 70-70 6c 69 63 61 74 69 6f  Type: applicatio 00000070: 6e 2f 6f 63 74 65 74 2d-73 74 72 65 61 6d 0d 0a  n/octet-stream.. 00000080: 43 6f 6e 74 65 6e 74 2d-4c 65 6e 67 74 68 3a 20  Content-Length: 00000090: 31 35 32 20 20 20 20 20-20 20 0d 0a 0d 0a 42 00  152      ....B. 000000a0: 15 32 78 01 00 00 00 90-42 00 77 01 00 00 00 48  .2x.....B.w....H 000000b0: 42 00 69 01 00 00 00 20-42 00 6a 02 00 00 00 04  B.i.... B.j..... 000000c0: 00 00 00 01 00 00 00 00-42 00 6b 02 00 00 00 04  .....B.k..... 000000d0: 00 00 00 01 00 00 00 00-42 00 50 02 00 00 00 04  .....B.P..... 000000e0: 00 00 01 00 00 00 00 00-42 00 0d 02 00 00 00 04  .....B..... 000000f0: 00 00 00 01 00 00 00 00-42 00 0f 01 00 00 00 38  .....B.....8 00000100: 42 00 5c 05 00 00 00 04-00 00 00 18 00 00 00 00  B.\..... 00000110: 42 00 79 01 00 00 00 20-42 00 74 05 00 00 00 04  B.y.... B.t..... 00000120: 00 00 00 01 00 00 00 00-42 00 74 05 00 00 00 04  .....B.t..... 00000130: 00 00 00 02 00 00 00 00-          ..... </pre>
0018 0019 0020 0021 0022 0023 0024 0025 0026 0027 0028 0029 0030 0031 0032	<pre> &lt;ResponseMessage&gt;   &lt;ResponseHeader&gt;     &lt;ProtocolVersion&gt;       &lt;ProtocolVersionMajor type="Integer" value="1"/&gt;       &lt;ProtocolVersionMinor type="Integer" value="1"/&gt;     &lt;/ProtocolVersion&gt;     &lt;TimeStamp type="DateTime" value="2014-06-10T08:03:34+00:00"/&gt;     &lt;BatchCount type="Integer" value="1"/&gt;   &lt;/ResponseHeader&gt; &lt;BatchItem&gt;     &lt;Operation type="Enumeration" value="Query"/&gt;     &lt;ResultStatus type="Enumeration" value="OperationFailed"/&gt;     &lt;ResultReason type="Enumeration" value="ResponseTooLarge"/&gt;     &lt;ResultMessage type="TextString" value="TOO_LARGE"/&gt;   &lt;/BatchItem&gt; &lt;/ResponseMessage&gt; </pre> <p>42007b01000000a042007a0100000048420069010000002042006a02000000040000000100000000  42006b020000000400000001000000004200920900000008000000005396bc2442000d0200000004  000000010000000042000f010000004842005c0500000004000000180000000042007f0500000004  000000010000000042007e0500000004000000020000000042007d0700000009544f4f5f4c415247  4500000000000000</p> <pre> 00000000: 48 54 54 50 2f 31 2e 31-20 32 30 30 20 4f 4b 0d  HTTP/1.1 200 OK. 00000010: 0a 43 6f 6e 74 65 6e 74-2d 54 79 70 65 3a 20 61  .Content-Type: a 00000020: 70 70 6c 69 63 61 74 69-6f 6e 2f 6f 63 74 65 74  pplication/octet 00000030: 2d 73 74 72 65 61 6d 0d-0a 43 6f 6e 74 65 6e 74  -stream.Content 00000040: 2d 4c 65 6e 67 74 68 3a-20 31 36 38 0d 0a 0d 0a  -Length: 168.... 00000050: 42 00 7b 01 00 00 00 a0-42 00 7a 01 00 00 00 48  B.{.... B.z....H 00000060: 42 00 69 01 00 00 00 20-42 00 6a 02 00 00 00 04  B.i.... B.j..... 00000070: 00 00 00 01 00 00 00 00-42 00 6b 02 00 00 00 04  .....B.k..... 00000080: 00 00 00 01 00 00 00 00-42 00 92 09 00 00 00 08  .....B..... 00000090: 00 00 00 00 53 96 bc 24-42 00 0d 02 00 00 00 04  ....S.&lt;\$B..... 000000a0: 00 00 00 01 00 00 00 00-42 00 0f 01 00 00 00 48  .....B.....H 000000b0: 42 00 5c 05 00 00 00 04-00 00 00 18 00 00 00 00  B.\..... 000000c0: 42 00 7f 05 00 00 00 04-00 00 00 01 00 00 00 00  B..... 000000d0: 42 00 7e 05 00 00 00 04-00 00 00 02 00 00 00 00  B.~..... 000000e0: 42 00 7d 07 00 00 00 09-54 4f 4f 5f 4c 41 52 47  B.}....TOO_LARG 000000f0: 45 00 00 00 00 00 00 00-          E..... </pre>
0033 0034 0035 0036 0037 0038 0039 0040 0041 0042	<pre> # TIME 1 &lt;RequestMessage&gt;   &lt;RequestHeader&gt;     &lt;ProtocolVersion&gt;       &lt;ProtocolVersionMajor type="Integer" value="1"/&gt;       &lt;ProtocolVersionMinor type="Integer" value="1"/&gt;     &lt;/ProtocolVersion&gt;     &lt;MaximumResponseSize type="Integer" value="2048"/&gt;     &lt;BatchCount type="Integer" value="1"/&gt;   &lt;/RequestHeader&gt;   &lt;BatchItem&gt; </pre>

0043	<Operation type="Enumeration" value="Query"/>
0044	<RequestPayload>
0045	<QueryFunction type="Enumeration" value="QueryOperations"/>
0046	<QueryFunction type="Enumeration" value="QueryObjects"/>
0047	</RequestPayload>
0048	</BatchItem>
0049	</RequestMessage>
	<p>42007801000000904200770100000048420069010000002042006a02000000040000000100000000</p> <p>42006b020000000400000001000000004200500200000004000008000000000042000d0200000004</p> <p>000000010000000042000f010000003842005c050000000400000018000000004200790100000020</p> <p>4200740500000004000000010000000042007405000000040000000200000000</p>
	<p>00000000: 50 4f 53 54 20 2f 6b 6d-69 70 20 48 54 54 50 2f POST /kmip HTTP/</p> <p>00000010: 31 2e 30 0d 0a 50 72 61-67 6d 61 3a 20 6e 6f 2d 1.0..Pragma: no-</p> <p>00000020: 63 61 63 68 65 0d 0a 43-61 63 68 65 2d 43 6f 6e cache..Cache-Con</p> <p>00000030: 74 72 6f 6c 3a 20 6e 6f-2d 63 61 63 68 65 0d 0a trol: no-cache..</p> <p>00000040: 43 6f 6e 6e 65 63 74 69-6f 6e 3a 20 6b 65 65 70 Connection: keep</p> <p>00000050: 2d 61 6c 69 76 65 0d 0a-43 6f 6e 74 65 6e 74 2d -alive..Content-</p> <p>00000060: 54 79 70 65 3a 20 61 70-70 6c 69 63 61 74 69 6f Type: applicatio</p> <p>00000070: 6e 2f 6f 63 74 65 74 2d-73 74 72 65 61 6d 0d 0a n/octet-stream..</p> <p>00000080: 43 6f 6e 74 65 6e 74 2d-4c 65 6e 67 74 68 3a 20 Content-Length:</p> <p>00000090: 31 35 32 20 20 20 20 20-20 20 0d 0a 0d 0a 42 00 152 ....B.</p> <p>000000a0: 15 32 78 01 00 00 00 90-42 00 77 01 00 00 00 48 .2x.....B.w....H</p> <p>000000b0: 42 00 69 01 00 00 00 20-42 00 6a 02 00 00 00 04 B.i.... B.j.....</p> <p>000000c0: 00 00 00 01 00 00 00 00-42 00 6b 02 00 00 00 04 .....B.k.....</p> <p>000000d0: 00 00 00 01 00 00 00 00-42 00 50 02 00 00 00 04 .....B.P.....</p> <p>000000e0: 00 00 08 00 00 00 00 00-42 00 0d 02 00 00 00 04 .....B.....</p> <p>000000f0: 00 00 00 01 00 00 00 00-42 00 0f 01 00 00 00 38 .....B.....8</p> <p>00000100: 42 00 5c 05 00 00 00 04-00 00 18 00 00 00 00 B.\.....</p> <p>00000110: 42 00 79 01 00 00 00 20-42 00 74 05 00 00 00 04 B.y.... B.t.....</p> <p>00000120: 00 00 00 01 00 00 00 00-42 00 74 05 00 00 00 04 .....B.t.....</p> <p>00000130: 00 00 00 02 00 00 00 00- ..... .....</p>
0050	<ResponseMessage>
0051	<ResponseHeader>
0052	<ProtocolVersion>
0053	<ProtocolVersionMajor type="Integer" value="1"/>
0054	<ProtocolVersionMinor type="Integer" value="1"/>
0055	</ProtocolVersion>
0056	<TimeStamp type="DateTime" value="2014-06-10T08:03:34+00:00"/>
0057	<BatchCount type="Integer" value="1"/>
0058	</ResponseHeader>
0059	<BatchItem>
0060	<Operation type="Enumeration" value="Query"/>
0061	<ResultStatus type="Enumeration" value="Success"/>
0062	<ResponsePayload>
0063	<Operation type="Enumeration" value="Query"/>
0064	<Operation type="Enumeration" value="Locate"/>
0065	<Operation type="Enumeration" value="Destroy"/>
0066	<Operation type="Enumeration" value="Get"/>
0067	<Operation type="Enumeration" value="Create"/>
0068	<Operation type="Enumeration" value="Register"/>
0069	<Operation type="Enumeration" value="GetAttributes"/>
0070	<Operation type="Enumeration" value="GetAttributeList"/>
0071	<Operation type="Enumeration" value="AddAttribute"/>
0072	<Operation type="Enumeration" value="ModifyAttribute"/>
0073	<Operation type="Enumeration" value="DeleteAttribute"/>
0074	<Operation type="Enumeration" value="Activate"/>
0075	<Operation type="Enumeration" value="Revoke"/>
0076	<Operation type="Enumeration" value="Poll"/>
0077	<Operation type="Enumeration" value="Cancel"/>
0078	<Operation type="Enumeration" value="Check"/>



0079	<Operation type="Enumeration" value="GetUsageAllocation"/>
0080	<Operation type="Enumeration" value="CreateKeyPair"/>
0081	<Operation type="Enumeration" value="ReKey"/>
0082	<Operation type="Enumeration" value="Archive"/>
0083	<Operation type="Enumeration" value="Recover"/>
0084	<Operation type="Enumeration" value="ObtainLease"/>
0085	<Operation type="Enumeration" value="ReKeyKeyPair"/>
0086	<Operation type="Enumeration" value="Certify"/>
0087	<Operation type="Enumeration" value="ReCertify"/>
0088	<Operation type="Enumeration" value="DiscoverVersions"/>
0089	<Operation type="Enumeration" value="Notify"/>
0090	<Operation type="Enumeration" value="Put"/>
0091	<ObjectType type="Enumeration" value="Certificate"/>
0092	<ObjectType type="Enumeration" value="SymmetricKey"/>
0093	<ObjectType type="Enumeration" value="SecretData"/>
0094	<ObjectType type="Enumeration" value="PublicKey"/>
0095	<ObjectType type="Enumeration" value="PrivateKey"/>
0096	<ObjectType type="Enumeration" value="Template"/>
0097	<ObjectType type="Enumeration" value="OpaqueObject"/>
0098	<ObjectType type="Enumeration" value="SplitKey"/>
0099	</ResponsePayload>
0100	</BatchItem>
0101	</ResponseMessage>
42007b01000002c042007a0100000048420069010000002042006a02000000040000000100000000 42006b020000000400000001000000004200920900000008000000005396bc2442000d0200000004 000000010000000042000f010000026842005c0500000004000000180000000042007f0500000004 000000000000000042007c010000024042005c0500000004000000180000000042005c0500000004 000000080000000042005c0500000004000000140000000042005c05000000040000000a00000000 42005c0500000004000000010000000042005c0500000004000000030000000042005c0500000004 0000000b0000000042005c05000000040000000c0000000042005c05000000040000000d00000000 42005c05000000040000000e0000000042005c05000000040000000f0000000042005c0500000004 000000120000000042005c0500000004000000130000000042005c05000000040000001a00000000 42005c0500000004000000190000000042005c0500000004000000090000000042005c0500000004 000000110000000042005c0500000004000000020000000042005c05000000040000000400000000 42005c0500000004000000150000000042005c0500000004000000160000000042005c0500000004 000000100000000042005c05000000040000001d0000000042005c05000000040000000600000000 42005c0500000004000000070000000042005c05000000040000001e0000000042005c0500000004 0000001b0000000042005c05000000040000001c0000000042005705000000040000000100000000 42005705000000040000000200000000420057050000000400000007000000004200570500000004 00000003000000004200570500000004000000040000000042005705000000040000000600000000 4200570500000004000000080000000042005705000000040000000500000000	
00000000: 48 54 54 50 2f 31 2e 31-20 32 30 30 20 4f 4b 0d HTTP/1.1 200 OK. 00000010: 0a 43 6f 6e 74 65 6e 74-2d 54 79 70 65 3a 20 61 .Content-Type: a 00000020: 70 70 6c 69 63 61 74 69-6f 6e 2f 6f 63 74 65 74 pplication/octet 00000030: 2d 73 74 72 65 61 6d 0d-0a 43 6f 6e 74 65 6e 74 -stream..Content 00000040: 2d 4c 65 6e 67 74 68 3a-20 37 31 32 0d 0a 0d 0a -Length: 712.... 00000050: 42 00 7b 01 00 00 02 c0-42 00 7a 01 00 00 00 48 B.{....@B.z....H 00000060: 42 00 69 01 00 00 00 20-42 00 6a 02 00 00 00 04 B.i.... B.j..... 00000070: 00 00 00 01 00 00 00 00-42 00 6b 02 00 00 00 04 .....B.k..... 00000080: 00 00 00 01 00 00 00 00-42 00 92 09 00 00 00 08 .....B..... 00000090: 00 00 00 00 53 96 bc 24-42 00 0d 02 00 00 00 04 ....S.<\$B..... 000000a0: 00 00 00 01 00 00 00 00-42 00 0f 01 00 00 02 68 .....B.....h 000000b0: 42 00 5c 05 00 00 00 04-00 00 18 00 00 00 00 B.\..... 000000c0: 42 00 7f 05 00 00 00 04-00 00 00 00 00 00 00 B..... 000000d0: 42 00 7c 01 00 00 02 40-42 00 5c 05 00 00 00 04 B. ....@B.\..... 000000e0: 00 00 00 18 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 000000f0: 00 00 00 08 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 00000100: 00 00 00 14 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 00000110: 00 00 00 0a 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 00000120: 00 00 00 01 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 00000130: 00 00 00 03 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 00000140: 00 00 00 0b 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 00000150: 00 00 00 0c 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\..... 00000160: 00 00 00 0d 00 00 00 00-42 00 5c 05 00 00 00 04 .....B.\.....	

00000170:	00 00 00 0e 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000180:	00 00 00 0f 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000190:	00 00 00 12 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000001a0:	00 00 00 13 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000001b0:	00 00 00 1a 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000001c0:	00 00 00 19 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000001d0:	00 00 00 09 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000001e0:	00 00 00 11 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000001f0:	00 00 00 02 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000200:	00 00 00 04 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000210:	00 00 00 15 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000220:	00 00 00 16 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000230:	00 00 00 10 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000240:	00 00 00 1d 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000250:	00 00 00 06 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000260:	00 00 00 07 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000270:	00 00 00 1e 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000280:	00 00 00 1b 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000290:	00 00 00 1c 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
000002a0:	00 00 00 01 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
000002b0:	00 00 00 02 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
000002c0:	00 00 00 07 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
000002d0:	00 00 00 03 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
000002e0:	00 00 00 04 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
000002f0:	00 00 00 06 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
00000300:	00 00 00 08 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
00000310:	00 00 00 05 00 00 00 00-	.....

122

## 123 3.3 Mandatory HTTPS Profile Test Cases KMIP v1.2

### 124 3.3.1 MSGENC-HTTPS-M-1-12 - Query, Maximum Response Size

125 Perform a Query operation, querying the Operations and Objects supported by the server, with a  
126 restriction on the Maximum Response Size set in the request header. Since the resulting Query response  
127 is too big, an error is returned. Increase the Maximum Response Size, resubmit the Query request, and  
128 get a successful response.

129 The specific list of operations and object types returned in the response MAY vary.

	# TIME 0
0001	<RequestMessage>
0002	<RequestHeader>
0003	<ProtocolVersion>
0004	<ProtocolVersionMajor type="Integer" value="1"/>
0005	<ProtocolVersionMinor type="Integer" value="2"/>
0006	</ProtocolVersion>
0007	<MaximumResponseSize type="Integer" value="256"/>
0008	<BatchCount type="Integer" value="1"/>
0009	</RequestHeader>
0010	<BatchItem>
0011	<Operation type="Enumeration" value="Query"/>
0012	<RequestPayload>
0013	<QueryFunction type="Enumeration" value="QueryOperations"/>
0014	<QueryFunction type="Enumeration" value="QueryObjects"/>
0015	</RequestPayload>
0016	</BatchItem>
0017	</RequestMessage>
	42007801000000904200770100000048420069010000002042006a02000000040000000100000000
	42006b0200000004000000002000000004200500200000004000001000000000042000d0200000004
	000000010000000042000f010000003842005c050000000400000018000000004200790100000020
	4200740500000004000000010000000042007405000000040000000200000000

	<pre> 00000000: 50 4f 53 54 20 2f 6b 6d-69 70 20 48 54 54 50 2f POST /kmip HTTP/ 00000010: 31 2e 30 0d 0a 50 72 61-67 6d 61 3a 20 6e 6f 2d 1.0..Pragma: no- 00000020: 63 61 63 68 65 0d 0a 43-61 63 68 65 2d 43 6f 6e cache..Cache-Con 00000030: 74 72 6f 6c 3a 20 6e 6f-2d 63 61 63 68 65 0d 0a trol: no-cache.. 00000040: 43 6f 6e 6e 65 63 74 69-6f 6e 3a 20 6b 65 65 70 Connection: keep 00000050: 2d 61 6c 69 76 65 0d 0a-43 6f 6e 74 65 6e 74 2d -alive..Content- 00000060: 54 79 70 65 3a 20 61 70-70 6c 69 63 61 74 69 6f Type: applicatio 00000070: 6e 2f 6f 63 74 65 74 2d-73 74 72 65 61 6d 0d 0a n/octet-stream.. 00000080: 43 6f 6e 74 65 6e 74 2d-4c 65 6e 67 74 68 3a 20 Content-Length: 00000090: 31 35 32 20 20 20 20 20-20 20 0d 0a 0d 0a 42 00 152 .....B. 000000a0: 15 32 78 01 00 00 00 90-42 00 77 01 00 00 00 48 .2x.....B.w....H 000000b0: 42 00 69 01 00 00 00 20-42 00 6a 02 00 00 00 04 B.i.... B.j.... 000000c0: 00 00 00 01 00 00 00 00-42 00 6b 02 00 00 00 04 .....B.k.... 000000d0: 00 00 00 02 00 00 00 00-42 00 50 02 00 00 00 04 .....B.P.... 000000e0: 00 00 01 00 00 00 00 00-42 00 0d 02 00 00 00 04 .....B..... 000000f0: 00 00 00 01 00 00 00 00-42 00 0f 01 00 00 00 38 .....B.....8 00000100: 42 00 5c 05 00 00 00 04-00 00 00 18 00 00 00 00 B.\..... 00000110: 42 00 79 01 00 00 00 20-42 00 74 05 00 00 00 04 B.y.... B.t.... 00000120: 00 00 00 01 00 00 00 00-42 00 74 05 00 00 00 04 .....B.t.... 00000130: 00 00 00 02 00 00 00 00- </pre>
0018	<pre> &lt;ResponseMessage&gt;   &lt;ResponseHeader&gt;     &lt;ProtocolVersion&gt;       &lt;ProtocolVersionMajor type="Integer" value="1"/&gt;       &lt;ProtocolVersionMinor type="Integer" value="2"/&gt;     &lt;/ProtocolVersion&gt;     &lt;TimeStamp type="DateTime" value="2014-06-10T08:07:28+00:00"/&gt;     &lt;BatchCount type="Integer" value="1"/&gt;   &lt;/ResponseHeader&gt; &lt;BatchItem&gt;     &lt;Operation type="Enumeration" value="Query"/&gt;     &lt;ResultStatus type="Enumeration" value="OperationFailed"/&gt;     &lt;ResultReason type="Enumeration" value="ResponseTooLarge"/&gt;     &lt;ResultMessage type="TextString" value="TOO_LARGE"/&gt;   &lt;/BatchItem&gt; &lt;/ResponseMessage&gt;  42007b010000000a042007a0100000048420069010000002042006a02000000040000000100000000 42006b0200000004000000002000000004200920900000008000000005396bcc042000d0200000004 0000000100000000042000f010000004842005c0500000004000000180000000042007f0500000004 0000000100000000042007e050000000400000020000000042007d0700000009544f4f5f4c415247 4500000000000000  00000000: 48 54 54 50 2f 31 2e 31-20 32 30 30 20 4f 4b 0d HTTP/1.1 200 OK. 00000010: 0a 43 6f 6e 74 65 6e 74-2d 54 79 70 65 3a 20 61 .Content-Type: a 00000020: 70 70 6c 69 63 61 74 69-6f 6e 2f 6f 63 74 65 74 pplication/octet 00000030: 2d 73 74 72 65 61 6d 0d-0a 43 6f 6e 74 65 6e 74 -stream..Content 00000040: 2d 4c 65 6e 67 74 68 3a-20 31 36 38 0d 0a 0d 0a -Length: 168.... 00000050: 42 00 7b 01 00 00 00 a0-42 00 7a 01 00 00 00 48 B.{.... B.z....H 00000060: 42 00 69 01 00 00 00 20-42 00 6a 02 00 00 00 04 B.i.... B.j.... 00000070: 00 00 00 01 00 00 00 00-42 00 6b 02 00 00 00 04 .....B.k.... 00000080: 00 00 00 02 00 00 00 00-42 00 92 09 00 00 00 08 .....B..... 00000090: 00 00 00 00 53 96 bd 50-42 00 0d 02 00 00 00 04 ....S.=PB..... 000000a0: 00 00 00 01 00 00 00 00-42 00 0f 01 00 00 00 48 .....B.....H 000000b0: 42 00 5c 05 00 00 00 04-00 00 00 18 00 00 00 00 B.\..... 000000c0: 42 00 7f 05 00 00 00 04-00 00 00 01 00 00 00 00 B..... 000000d0: 42 00 7e 05 00 00 00 04-00 00 00 02 00 00 00 00 B.~..... 000000e0: 42 00 7d 07 00 00 00 09-54 4f 4f 5f 4c 41 52 47 B.}.....TOO_LARG 000000f0: 45 00 00 00 00 00 00 00- E..... </pre>
0033	
0034	
0035	
0036	
0037	
	<pre> # TIME 1 &lt;RequestMessage&gt;   &lt;RequestHeader&gt;     &lt;ProtocolVersion&gt;       &lt;ProtocolVersionMajor type="Integer" value="1"/&gt;       &lt;ProtocolVersionMinor type="Integer" value="2"/&gt; </pre>

0038	</ProtocolVersion>
0039	<MaximumResponseSize type="Integer" value="2048"/>
0040	<BatchCount type="Integer" value="1"/>
0041	</RequestHeader>
0042	<BatchItem>
0043	<Operation type="Enumeration" value="Query"/>
0044	<RequestPayload>
0045	<QueryFunction type="Enumeration" value="QueryOperations"/>
0046	<QueryFunction type="Enumeration" value="QueryObjects"/>
0047	</RequestPayload>
0048	</BatchItem>
0049	</RequestMessage>
	<p>420078010000000904200770100000048420069010000002042006a02000000040000000100000000</p> <p>42006b0200000004000000002000000004200500200000004000008000000000042000d0200000004</p> <p>0000000100000000042000f010000003842005c050000000400000018000000004200790100000020</p> <p>4200740500000004000000010000000042007405000000040000000200000000</p> <p>00000000: 50 4f 53 54 20 2f 6b 6d-69 70 20 48 54 54 50 2f POST /kmp HTTP/</p> <p>00000010: 31 2e 30 0d 0a 50 72 61-67 6d 61 3a 20 6e 6f 2d 1.0..Pragma: no-</p> <p>00000020: 63 61 63 68 65 0d 0a 43-61 63 68 65 2d 43 6f 6e cache..Cache-Con</p> <p>00000030: 74 72 6f 6c 3a 20 6e 6f-2d 63 61 63 68 65 0d 0a trol: no-cache..</p> <p>00000040: 43 6f 6e 6e 65 63 74 69-6f 6e 3a 20 6b 65 65 70 Connection: keep</p> <p>00000050: 2d 61 6c 69 76 65 0d 0a-43 6f 6e 74 65 6e 74 2d -alive..Content-</p> <p>00000060: 54 79 70 65 3a 20 61 70-70 6c 69 63 61 74 69 6f Type: applicatio</p> <p>00000070: 6e 2f 6f 63 74 65 74 2d-73 74 72 65 61 6d 0d 0a n/octet-stream..</p> <p>00000080: 43 6f 6e 74 65 6e 74 2d-4c 65 6e 67 74 68 3a 20 Content-Length:</p> <p>00000090: 31 35 32 20 20 20 20 20-20 20 0d 0a 0d 0a 42 00 152 .....B.</p> <p>000000a0: 15 32 78 01 00 00 00 90-42 00 77 01 00 00 00 48 .2x.....B.w....H</p> <p>000000b0: 42 00 69 01 00 00 00 20-42 00 6a 02 00 00 00 04 B.i.... B.j....</p> <p>000000c0: 00 00 00 01 00 00 00 00-42 00 6b 02 00 00 00 04 .....B.k....</p> <p>000000d0: 00 00 00 02 00 00 00 00-42 00 50 02 00 00 00 04 .....B.P....</p> <p>000000e0: 00 00 08 00 00 00 00 00-42 00 0d 02 00 00 00 04 .....B.....</p> <p>000000f0: 00 00 00 01 00 00 00 00-42 00 0f 01 00 00 00 38 .....B.....8</p> <p>00000100: 42 00 5c 05 00 00 00 04-00 00 00 18 00 00 00 00 B.\.....</p> <p>00000110: 42 00 79 01 00 00 00 20-42 00 74 05 00 00 00 04 B.y.... B.t....</p> <p>00000120: 00 00 00 01 00 00 00 00-42 00 74 05 00 00 00 04 .....B.t....</p> <p>00000130: 00 00 00 02 00 00 00 00-</p>
0050	<ResponseMessage>
0051	<ResponseHeader>
0052	<ProtocolVersion>
0053	<ProtocolVersionMajor type="Integer" value="1"/>
0054	<ProtocolVersionMinor type="Integer" value="2"/>
0055	</ProtocolVersion>
0056	<TimeStamp type="DateTime" value="2014-06-10T08:07:28+00:00"/>
0057	<BatchCount type="Integer" value="1"/>
0058	</ResponseHeader>
0059	<BatchItem>
0060	<Operation type="Enumeration" value="Query"/>
0061	<ResultStatus type="Enumeration" value="Success"/>
0062	<ResponsePayload>
0063	<Operation type="Enumeration" value="Query"/>
0064	<Operation type="Enumeration" value="Locate"/>
0065	<Operation type="Enumeration" value="Destroy"/>
0066	<Operation type="Enumeration" value="Get"/>
0067	<Operation type="Enumeration" value="Create"/>
0068	<Operation type="Enumeration" value="Register"/>
0069	<Operation type="Enumeration" value="GetAttributes"/>
0070	<Operation type="Enumeration" value="GetAttributeList"/>
0071	<Operation type="Enumeration" value="AddAttribute"/>
0072	<Operation type="Enumeration" value="ModifyAttribute"/>
0073	<Operation type="Enumeration" value="DeleteAttribute"/>

0074	<Operation type="Enumeration" value="Activate"/>
0075	<Operation type="Enumeration" value="Revoke"/>
0076	<Operation type="Enumeration" value="Poll"/>
0077	<Operation type="Enumeration" value="Cancel"/>
0078	<Operation type="Enumeration" value="Check"/>
0079	<Operation type="Enumeration" value="GetUsageAllocation"/>
0080	<Operation type="Enumeration" value="CreateKeyPair"/>
0081	<Operation type="Enumeration" value="ReKey"/>
0082	<Operation type="Enumeration" value="Archive"/>
0083	<Operation type="Enumeration" value="Recover"/>
0084	<Operation type="Enumeration" value="ObtainLease"/>
0085	<Operation type="Enumeration" value="ReKeyKeyPair"/>
0086	<Operation type="Enumeration" value="Certify"/>
0087	<Operation type="Enumeration" value="ReCertify"/>
0088	<Operation type="Enumeration" value="DiscoverVersions"/>
0089	<Operation type="Enumeration" value="Notify"/>
0090	<Operation type="Enumeration" value="Put"/>
0091	<Operation type="Enumeration" value="RNGRetrieve"/>
0092	<Operation type="Enumeration" value="RNGSeed"/>
0093	<Operation type="Enumeration" value="Encrypt"/>
0094	<Operation type="Enumeration" value="Decrypt"/>
0095	<Operation type="Enumeration" value="Sign"/>
0096	<Operation type="Enumeration" value="SignatureVerify"/>
0097	<Operation type="Enumeration" value="MAC"/>
0098	<Operation type="Enumeration" value="MACVerify"/>
0099	<Operation type="Enumeration" value="Hash"/>
0100	<Operation type="Enumeration" value="CreateSplitKey"/>
0101	<Operation type="Enumeration" value="JoinSplitKey"/>
0102	<ObjectType type="Enumeration" value="Certificate"/>
0103	<ObjectType type="Enumeration" value="SymmetricKey"/>
0104	<ObjectType type="Enumeration" value="SecretData"/>
0105	<ObjectType type="Enumeration" value="PublicKey"/>
0106	<ObjectType type="Enumeration" value="PrivateKey"/>
0107	<ObjectType type="Enumeration" value="Template"/>
0108	<ObjectType type="Enumeration" value="OpaqueObject"/>
0109	<ObjectType type="Enumeration" value="SplitKey"/>
0110	<ObjectType type="Enumeration" value="PGPKey"/>
0111	</ResponsePayload>
0112	</BatchItem>
0113	</ResponseMessage>
	42007b010000038042007a0100000048420069010000002042006a02000000040000000100000000
	42006b0200000004000000002000000004200920900000008000000005396bcc042000d0200000004
	000000010000000042000f010000032842005c0500000004000000180000000042007f0500000004
	000000000000000042007c010000030042005c0500000004000000180000000042005c0500000004
	000000080000000042005c0500000004000000140000000042005c05000000040000000a00000000
	42005c0500000004000000010000000042005c0500000004000000030000000042005c0500000004
	0000000b0000000042005c05000000040000000c0000000042005c05000000040000000d00000000
	42005c05000000040000000e0000000042005c05000000040000000f0000000042005c0500000004
	000000120000000042005c0500000004000000130000000042005c05000000040000001a00000000
	42005c0500000004000000190000000042005c0500000004000000090000000042005c0500000004
	000000110000000042005c0500000004000000020000000042005c05000000040000000400000000
	42005c0500000004000000150000000042005c0500000004000000160000000042005c0500000004
	000000100000000042005c05000000040000001d0000000042005c05000000040000000600000000
	42005c0500000004000000070000000042005c05000000040000001e0000000042005c0500000004
	0000001b0000000042005c05000000040000001c0000000042005c05000000040000000250000000
	42005c0500000004000000260000000042005c05000000040000001f0000000042005c0500000004
	000000200000000042005c0500000004000000210000000042005c05000000040000002200000000
	42005c0500000004000000230000000042005c0500000004000000240000000042005c0500000004
	000000270000000042005c0500000004000000280000000042005c05000000040000002900000000
	42005705000000040000000100000000420057050000000400000002000000004200570500000004
	00000007000000004200570500000004000000030000000042005705000000040000000400000000

4200570500000004000000006000000004200570500000004000000080000000042005705000000040000000500000000042005705000000040000000900000000	
00000000: 48 54 54 50 2f 31 2e 31-20 32 30 30 20 4f 4b 0d	HTTP/1.1 200 OK.
00000010: 0a 43 6f 6e 74 65 6e 74-2d 54 79 70 65 3a 20 61	.Content-Type: a
00000020: 70 70 6c 69 63 61 74 69-6f 6e 2f 6f 63 74 65 74	pplication/octet
00000030: 2d 73 74 72 65 61 6d 0d-0a 43 6f 6e 74 65 6e 74	-stream..Content
00000040: 2d 4c 65 6e 67 74 68 3a-20 39 30 34 0d 0a 0d 0a	-Length: 904....
00000050: 42 00 7b 01 00 00 03 80-42 00 7a 01 00 00 00 48	B.{.....B.z....H
00000060: 42 00 69 01 00 00 00 20-42 00 6a 02 00 00 00 04	B.i..... B.j.....
00000070: 00 00 00 01 00 00 00 00-42 00 6b 02 00 00 00 04	.....B.k.....
00000080: 00 00 00 02 00 00 00 00-42 00 92 09 00 00 00 08	.....B.....
00000090: 00 00 00 00 53 96 bd 50-42 00 0d 02 00 00 00 04	....S.=PB.....
000000a0: 00 00 00 01 00 00 00 00-42 00 0f 01 00 00 03 28	.....B..... (
000000b0: 42 00 5c 05 00 00 00 04-00 00 00 18 00 00 00 00	B.\.....
000000c0: 42 00 7f 05 00 00 00 04-00 00 00 00 00 00 00 00	B.....
000000d0: 42 00 7c 01 00 00 03 00-42 00 5c 05 00 00 00 04	B. .....B.\.....
000000e0: 00 00 00 18 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000000f0: 00 00 00 08 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000100: 00 00 00 14 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000110: 00 00 00 0a 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000120: 00 00 00 01 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000130: 00 00 00 03 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000140: 00 00 00 0b 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000150: 00 00 00 0c 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000160: 00 00 00 0d 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000170: 00 00 00 0e 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000180: 00 00 00 0f 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000190: 00 00 00 12 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000001a0: 00 00 00 13 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000001b0: 00 00 00 1a 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000001c0: 00 00 00 19 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000001d0: 00 00 00 09 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000001e0: 00 00 00 11 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000001f0: 00 00 00 02 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000200: 00 00 00 04 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000210: 00 00 00 15 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000220: 00 00 00 16 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000230: 00 00 00 10 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000240: 00 00 00 1d 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000250: 00 00 00 06 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000260: 00 00 00 07 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000270: 00 00 00 1e 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000280: 00 00 00 1b 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
00000290: 00 00 00 1c 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000002a0: 00 00 00 25 00 00 00 00-42 00 5c 05 00 00 00 04	...%.B.\.....
000002b0: 00 00 00 26 00 00 00 00-42 00 5c 05 00 00 00 04	...&...B.\.....
000002c0: 00 00 00 1f 00 00 00 00-42 00 5c 05 00 00 00 04	.....B.\.....
000002d0: 00 00 00 20 00 00 00 00-42 00 5c 05 00 00 00 04	... ..B.\.....
000002e0: 00 00 00 21 00 00 00 00-42 00 5c 05 00 00 00 04	...!.B.\.....
000002f0: 00 00 00 22 00 00 00 00-42 00 5c 05 00 00 00 04	...".B.\.....
00000300: 00 00 00 23 00 00 00 00-42 00 5c 05 00 00 00 04	...#.B.\.....
00000310: 00 00 00 24 00 00 00 00-42 00 5c 05 00 00 00 04	...\$.B.\.....
00000320: 00 00 00 27 00 00 00 00-42 00 5c 05 00 00 00 04	...'.B.\.....
00000330: 00 00 00 28 00 00 00 00-42 00 5c 05 00 00 00 04	... (.B.\.....
00000340: 00 00 00 29 00 00 00 00-42 00 57 05 00 00 00 04	...).B.W.....
00000350: 00 00 00 01 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
00000360: 00 00 00 02 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
00000370: 00 00 00 07 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
00000380: 00 00 00 03 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
00000390: 00 00 00 04 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
000003a0: 00 00 00 06 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
000003b0: 00 00 00 08 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
000003c0: 00 00 00 05 00 00 00 00-42 00 57 05 00 00 00 04	.....B.W.....
000003d0: 00 00 00 09 00 00 00 00-	.....

---

## 4 JSON Profile

The JSON profile specifies the use of KMIP replacing the TTLV message encoding with a JSON message encoding. The results returned using the JSON encoding SHALL be logically the same as if the message encoding was in TTLV form. All size or length values specified within tag values for KMIP items SHALL be the same in JSON form as if the message encoding were in TTLV form. The implications of this are that items such as MaximumResponseSize are interpreted to refer to a maximum length computed as if it were a TTLV-encoded response, not the length of the JSON-encoded response.

### 4.1 JSON Encoding

#### 4.1.1 Hex representations

Hex representations of numbers must always begin with '0x' and must not include any spaces. They may use either upper or lower case 'a'-'f'. The hex representation must include all leading zeros or sign extension bits when representing a value of a fixed width such as Tags (3 bytes), Integer (32-bit signed big-endian), Long Integer (64-bit signed big-endian) and Big Integer (big-endian multiple of 8 bytes). The Integer values for -1, 0, 1 are represented as "0xffffffff", "0x00000000", "0x00000001". Hex representation for Byte Strings are similar to numbers, but do not include the '0x' prefix, and can be of any length.

#### 4.1.2 Tags

Tags are a String that may contain either:

- The 3-byte tag hex value prefixed with '0x'
- The normalised text of a Tag as specified in the KMIP Specification

Other text values may be used such as published names of Extension tags, or names of new tags added in future KMIP versions. Producers may however choose to use hex values for these tags to ensure they are understood by all consumers.

#### 4.1.3 Normalizing Names

KMIP text values of Tags, Types and Enumerations SHALL be normalized to create a 'CamelCase' format that would be suitable to be used as a variable name in C/Java or an JSON name.

The basic approach to converting from KMIP text to CamelCase is to separate the text into individual word tokens (rules 1-4), capitalize the first letter of each word (rule 5) and then join with spaces removed (rule 6). The tokenizing splits on whitespace and on dashes where the token following is a valid word. The tokenizing also removes round brackets and shifts decimals from the front to the back of the first word in each string. The following rules SHALL be applied to create the normalized CamelCase form:

1. Replace round brackets '(', ')' with spaces
2. If a non-word char (not alpha, digit or underscore) is followed by a letter (either upper or lower case) then a lower case letter, replace the non-word char with space
3. Replace remaining non-word chars (except whitespace) with underscore.
4. If the first word begins with a digit, move all digits at start of first word to end of first word
5. Capitalize the first letter of each word
6. Concatenate all words with spaces removed

```

170 # 1. Replace brackets with space
171 noBrackets = re.sub('[\(\)]', ' ', enumName)
172 # 2. replace \W with space if followed by letter, lower
173 nonWordToSpace = re.sub('\W([A-Za-z][a-z])', r' \1', noBrackets)
174 # 3. non-word to underscore
175 words = [re.sub('\W', '_', s) for s in nonWordToSpace.split()]
176 # 4. move numbers to end of first word
177 words[0] = re.sub('^(\\d+)(.*)', r'\\2\\1', words[0])
178 # 5. captialize first letter of each word
179 words = [re.sub('^.\\.', s[0].upper(), s) for s in words]
180 # 6. concatenate
181 enumNameCamel = ''.join(words)

```

*Example python name normalization code*

```

184 # 1. Replace brackets with space
185 $enumName=~s/[\\(\\)]/ /g;
186 # 2. replace \W with space if followed by letter, lower
187 $enumName=~s/\\W([A-Za-z][a-z])/ \\1/g;
188 # 3. non-word to underscore
189 @words=split(/ /,$enumName);
190 for($i=0;$i<=$#words;$i++) { $words[$i]=~s/\\W/_/g; }
191 # 4. move numbers to end of first word
192 $words[0] =~ s/^(\\d+)(.*)/\\2\\1/;
193 # 5. captialize first letter of each word
194 for($i=0;$i<=$#words;$i++) {
195     substr($words[$i],0,1)=~tr/a-z/A-Z/;
196 }
197 # 6. concatenate
198 $enumNameCamel = join('',$words);
199

```

*Example perl name normalization code*

#### 4.1.4 Type

Type must be a String containing one of the normalized CamelCase values as defined in the KMIP specification.

- Structure
- Integer
- LongInteger
- BigInteger
- Enumeration
- Boolean
- TextString
- ByteString
- DateTime
- Interval

If type is not included, the default type of Structure SHALL be used.

#### 4.1.5 Value

The specification of a value is represented differently for each TTLV type.



## 4.1.6 JSON Object

For JSON encoding, each TTLV is represented as a JSON Object with properties 'tag', optional 'name', 'type' and 'value'.

```
{"tag": "ActivationDate", "type": "DateTime", "value": "2001-01-01T10:00:00+10:00"}
{"tag": "0x54FFFF", "name": "SomeExtension", "type": "Integer", "value": "0x00000001"}
```

The 'type' property / attribute SHALL have a default value of 'Structure' and may be omitted for Structures.

### 4.1.6.1 Tags

Tags are a String that may contain either:

- The 3-byte tag hex value prefixed with '0x'
- The normalised text of a Tag as specified in the KMIP Specification

Other text values may be used such as published names of Extension tags, or names of new tags added in future KMIP versions. Producers may however choose to use hex values for these tags to ensure they are understood by all consumers.

```
{"tag": "0x420001", "type": "DateTime", "value": "2001-01-01T10:00:00+10:00"}
{"tag": "ActivationDate", "type": "DateTime", "value": "2001-01-01T10:00:00+10:00"}
{"tag": "IVCounterNonce", "type": "ByteString", "value": "alb2c3d4"}
{"tag": "PrivateKeyTemplateAttribute", "type": "Structure", "value": []}
{"tag": "0x545352", "type": "TextString", "value": "This is an extension"}
{"tag": "WELL_KNOWN_EXTENSION", "type": "TextString", "value": "This is an extension"}
```

### 4.1.6.2 Structure

For JSON, value is an Array containing sub-items, or may be null.

```
{"tag": "ProtocolVersion", "type": "Structure", "value": [
  {"tag": "ProtocolVersionMajor", "type": "Integer", "value": 1},
  {"tag": "ProtocolVersionMinor", "type": "Integer", "value": 0}
]}
{"tag": "ProtocolVersion", "value": [
  {"tag": "ProtocolVersionMajor", "type": "Integer", "value": 1},
  {"tag": "ProtocolVersionMinor", "type": "Integer", "value": 0}
]}
```

The 'type' property / attribute is optional for a Structure.

### 4.1.6.3 Integer

For JSON, value is either a Number or a hex string.

```
{"tag": "BatchCount", "type": "Integer", "value": 10}
{"tag": "BatchCount", "type": "Integer", "value": "0x0000000A"}
```

### 4.1.6.4 Integer - Special case for Masks

(Cryptographic Usage Mask, Storage Status Mask):

Integer mask values can also be encoded as a String containing mask components. JSON uses '|' as the separator. Components may be either the text of the enumeration value as defined in the KMIP Specification or a 32-bit unsigned big-endian hex string.

```
{"tag": "CryptographicUsageMask", "type": "Integer", "value": "0x0000100c"}
{"tag": "CryptographicUsageMask", "type": "Integer", "value": "Encrypt|Decrypt|CertificateSign"}
{"tag": "CryptographicUsageMask", "type": "Integer", "value":
  "CertificateSign|0x00000004|0x00000008"}
{"tag": "CryptographicUsageMask", "type": "Integer", "value": "CertificateSign|0x0000000c"}
```

#### 4.1.6.5 Long Integer

For JSON, value is either a Number or a hex string. Note that JS Numbers are 64-bit floating point and can only represent 53-bits of precision, so any values  $\geq 2^{52}$  must be represented as hex strings.

```
{ "tag": "0x540001", "type": "LongInteger", "value": "0xfffffffffffffffe" }
{ "tag": "0x540001", "type": "LongInteger", "value": -2 }
{ "tag": "UsageLimitsCount", "type": "LongInteger", "value": "0x1000000000000000" }
```

Note that this value ( $2^{60}$ ) is too large to be represented as a Number in JSON.

#### 4.1.6.6 Big Integer

For JSON, value is either a Number or a hex string. Note that Big Integers must be sign extended to contain a multiple of 8 bytes, and as per LongInteger, JS numbers only support a limited range of values.

```
{ "tag": "X", "type": "BigInteger", "value": 0 }
{ "tag": "X", "type": "BigInteger", "value": "0x0000000000000000" }
```

#### 4.1.6.7 Enumeration

For JSON, value may contain:

- Number representing the enumeration 32-bit unsigned big-endian value
- Hex string representation of 32-bit unsigned big-endian value
- CamelCase enum text as defined in KMIP 9.1.3.2.x

```
{ "tag": "0x420057", "type": "Enumeration", "value": 2 }
{ "tag": "ObjectType", "type": "Enumeration", "value": "0x00000002" }
{ "tag": "ObjectType", "type": "Enumeration", "value": "SymmetricKey" }
```

#### 4.1.6.8 Boolean

For JSON, value must be either a hex string, or a JSON Boolean 'true' or 'false'.

```
{ "tag": "BatchOrderOption", "type": "Boolean", "value": true }
{ "tag": "BatchOrderOption", "type": "Boolean", "value": "0x0000000000000001" }
```

#### 4.1.6.9 Text String

For JSON, value must be a String

```
{ "tag": "AttributeName", "type": "TextString", "value": "Cryptographic Algorithm" }
```

#### 4.1.6.10 Byte String

For JSON, value must be a hex string. Note Byte Strings do not include the '0x' prefix, and do not have any leading bytes.

```
{ "tag": "MACSignature", "type": "ByteString", "value": "C50F77" }
```

#### 4.1.6.11 Date-Time

For JSON, value must be either a hex string, or an ISO8601 DateTime as used in XSD using format:

```
'-'? yyyy '-' mm '-' dd 'T' hh ':' mm ':' ss ('.' s)? ((( '+' | '-' ) hh ':' mm ) | 'Z')?
```

Fractional seconds are not used in KMIP and should not generally be shown. If they are used, they should be ignored (truncated).

```
{ "tag": "ArchiveDate", "type": "DateTime", "value": "0x000000003a505520" }
{ "tag": "ArchiveDate", "type": "DateTime", "value": "2001-01-01T10:00:00+10:00" }
```

#### 4.1.6.12 Interval

For JSON, value is either a Number or a hex string. Note that intervals are 32-bit unsigned big-endian values.

```
{ "tag": "Offset", "type": "Interval", "value": 27 }
{ "tag": "Offset", "type": "Interval", "value": "0x0000001b" }
```

## 5 JSON Profile Test Cases

The test cases define a number of request-response pairs for KMIP operations. Each test case is provided in the XML format specified in section 6 intended to be both human-readable and usable by automated tools. The time sequence (starting from 0) for each request-response pair is noted and line numbers are provided for ease of cross-reference for a given test sequence.

Each test case has a unique label (the section name) which includes indication of mandatory (-M-) or optional (-O-) status and the protocol version major and minor numbers as part of the identifier.

The test cases may depend on a specific configuration of a KMIP client and server being configured in a manner consistent with the test case assumptions.

Where possible the flow of unique identifiers between tests, the date-time values, and other dynamic items are indicated using symbolic identifiers – in actual request and response messages these dynamic values will be filled in with valid values.

Note: the values for the returned items and the custom attributes are illustrative. Actual values from a real client system may vary as specified in section 8.4

### 5.1 Mandatory JSON Profile Test Cases KMIP v1.0

#### 5.1.1 MSGENC-JSON-M-1-10 - Query, Maximum Response Size

Perform a Query operation, querying the Operations and Objects supported by the server, with a restriction on the Maximum Response Size set in the request header. Since the resulting Query response is too big, an error is returned. Increase the Maximum Response Size, resubmit the Query request, and get a successful response.

The specific list of operations and object types returned in the response MAY vary.

	<pre># TIME 0 0001 &lt;RequestMessage&gt; 0002   &lt;RequestHeader&gt; 0003     &lt;ProtocolVersion&gt; 0004       &lt;ProtocolVersionMajor type="Integer" value="1"/&gt; 0005       &lt;ProtocolVersionMinor type="Integer" value="0"/&gt; 0006     &lt;/ProtocolVersion&gt; 0007     &lt;MaximumResponseSize type="Integer" value="256"/&gt; 0008     &lt;BatchCount type="Integer" value="1"/&gt; 0009   &lt;/RequestHeader&gt; 0010   &lt;BatchItem&gt; 0011     &lt;Operation type="Enumeration" value="Query"/&gt; 0012     &lt;RequestPayload&gt; 0013       &lt;QueryFunction type="Enumeration" value="QueryOperations"/&gt; 0014       &lt;QueryFunction type="Enumeration" value="QueryObjects"/&gt; 0015     &lt;/RequestPayload&gt; 0016   &lt;/BatchItem&gt; 0017 &lt;/RequestMessage&gt;  42007801000000904200770100000048420069010000002042006a02000000040000000100000000 42006b02000000040000000000000004200500200000004000001000000000042000d0200000004 000000010000000042000f010000003842005c050000000400000018000000004200790100000020 4200740500000004000000010000000042007405000000040000000200000000  {"tag":"RequestMessage", "value":{   {"tag":"RequestHeader", "value":{     {"tag":"ProtocolVersion", "value":{       {"tag":"ProtocolVersionMajor", "type":"Integer", "value":"0x00000001"},       {"tag":"ProtocolVersionMinor", "type":"Integer", "value":"0x00000000"}     }   } }</pre>
--	--

	<pre>     }},     {"tag": "MaximumResponseSize", "type": "Integer", "value": "0x00000100"},     {"tag": "BatchCount", "type": "Integer", "value": "0x00000001"}   ]},   {"tag": "BatchItem", "value": [     {"tag": "Operation", "type": "Enumeration", "value": "Query"},     {"tag": "RequestPayload", "value": [       {"tag": "QueryFunction", "type": "Enumeration", "value": "QueryOperations"},       {"tag": "QueryFunction", "type": "Enumeration", "value": "QueryObjects"}     ]}   ]} ] </pre>
0018 0019 0020 0021 0022 0023 0024 0025 0026 0027 0028 0029 0030 0031 0032	<pre> &lt;ResponseMessage&gt;   &lt;ResponseHeader&gt;     &lt;ProtocolVersion&gt;       &lt;ProtocolVersionMajor type="Integer" value="1"/&gt;       &lt;ProtocolVersionMinor type="Integer" value="0"/&gt;     &lt;/ProtocolVersion&gt;     &lt;TimeStamp type="DateTime" value="2013-06-26T09:09:17+00:00"/&gt;     &lt;BatchCount type="Integer" value="1"/&gt;   &lt;/ResponseHeader&gt;   &lt;BatchItem&gt;     &lt;Operation type="Enumeration" value="Query"/&gt;     &lt;ResultStatus type="Enumeration" value="OperationFailed"/&gt;     &lt;ResultReason type="Enumeration" value="ResponseTooLarge"/&gt;     &lt;ResultMessage type="TextString" value="TOO_LARGE"/&gt;   &lt;/BatchItem&gt; &lt;/ResponseMessage&gt; </pre> <p>42007b01000000a042007a0100000048420069010000002042006a02000000040000000100000000 42006b020000000400000000000000042009209000000080000000051caaabd42000d0200000004 000000010000000042000f010000004842005c0500000004000000180000000042007f0500000004 000000010000000042007e0500000004000000020000000042007d0700000009544f4f5f4c415247 4500000000000000</p> <pre> {"tag": "ResponseMessage", "value": [   {"tag": "ResponseHeader", "value": [     {"tag": "ProtocolVersion", "value": [       {"tag": "ProtocolVersionMajor", "type": "Integer", "value": "0x00000001"},       {"tag": "ProtocolVersionMinor", "type": "Integer", "value": "0x00000000"}     ]},     {"tag": "TimeStamp", "type": "DateTime", "value": "2013-06-26T09:09:17+00:00"},     {"tag": "BatchCount", "type": "Integer", "value": "0x00000001"}   ]},   {"tag": "BatchItem", "value": [     {"tag": "Operation", "type": "Enumeration", "value": "Query"},     {"tag": "ResultStatus", "type": "Enumeration", "value": "OperationFailed"},     {"tag": "ResultReason", "type": "Enumeration", "value": "ResponseTooLarge"},     {"tag": "ResultMessage", "type": "TextString", "value": "TOO_LARGE"}   ]} ]} </pre>
0032 0033 0034 0035 0036 0037 0038 0039 0040 0041 0042	<pre> # TIME 1 &lt;RequestMessage&gt;   &lt;RequestHeader&gt;     &lt;ProtocolVersion&gt;       &lt;ProtocolVersionMajor type="Integer" value="1"/&gt;       &lt;ProtocolVersionMinor type="Integer" value="0"/&gt;     &lt;/ProtocolVersion&gt;     &lt;MaximumResponseSize type="Integer" value="2048"/&gt;     &lt;BatchCount type="Integer" value="1"/&gt;   &lt;/RequestHeader&gt;   &lt;BatchItem&gt;     &lt;Operation type="Enumeration" value="Query"/&gt; </pre>

0043	<RequestPayload>
0044	<QueryFunction type="Enumeration" value="QueryOperations"/>
0045	<QueryFunction type="Enumeration" value="QueryObjects"/>
0046	</RequestPayload>
0047	</BatchItem>
0048	</RequestMessage>
	<p>42007801000000904200770100000048420069010000002042006a02000000040000000100000000</p> <p>42006b02000000040000000000000000420050020000000400000080000000000042000d0200000004</p> <p>000000010000000042000f010000003842005c050000000400000018000000004200790100000020</p> <p>4200740500000004000000010000000042007405000000040000000200000000</p> <pre>{   "tag": "RequestMessage", "value": [     {       "tag": "RequestHeader", "value": [         {           "tag": "ProtocolVersion", "value": [             {               "tag": "ProtocolVersionMajor", "type": "Integer", "value": "0x00000001",             },             {               "tag": "ProtocolVersionMinor", "type": "Integer", "value": "0x00000000"             }           ],         },         {           "tag": "MaximumResponseSize", "type": "Integer", "value": "0x00000800",         },         {           "tag": "BatchCount", "type": "Integer", "value": "0x00000001"         }       ],     },     {       "tag": "BatchItem", "value": [         {           "tag": "Operation", "type": "Enumeration", "value": "Query",         },         {           "tag": "RequestPayload", "value": [             {               "tag": "QueryFunction", "type": "Enumeration", "value": "QueryOperations",             },             {               "tag": "QueryFunction", "type": "Enumeration", "value": "QueryObjects"             }           ]         }       ]     }   ] }</pre>
0049	<ResponseMessage>
0050	<ResponseHeader>
0051	<ProtocolVersion>
0052	<ProtocolVersionMajor type="Integer" value="1"/>
0053	<ProtocolVersionMinor type="Integer" value="0"/>
0054	</ProtocolVersion>
0055	<TimeStamp type="DateTime" value="2013-06-26T09:09:17+00:00"/>
0056	<BatchCount type="Integer" value="1"/>
0057	</ResponseHeader>
0058	<BatchItem>
0059	<Operation type="Enumeration" value="Query"/>
0060	<ResultStatus type="Enumeration" value="Success"/>
0061	<ResponsePayload>
0062	<Operation type="Enumeration" value="Query"/>
0063	<Operation type="Enumeration" value="Locate"/>
0064	<Operation type="Enumeration" value="Destroy"/>
0065	<Operation type="Enumeration" value="Get"/>
0066	<Operation type="Enumeration" value="Create"/>
0067	<Operation type="Enumeration" value="Register"/>
0068	<Operation type="Enumeration" value="GetAttributes"/>
0069	<Operation type="Enumeration" value="GetAttributeList"/>
0070	<Operation type="Enumeration" value="AddAttribute"/>
0071	<Operation type="Enumeration" value="ModifyAttribute"/>
0072	<Operation type="Enumeration" value="DeleteAttribute"/>
0073	<Operation type="Enumeration" value="Activate"/>
0074	<Operation type="Enumeration" value="Revoke"/>
0075	<Operation type="Enumeration" value="Poll"/>
0076	<Operation type="Enumeration" value="Cancel"/>
0077	<Operation type="Enumeration" value="Check"/>
0078	<Operation type="Enumeration" value="GetUsageAllocation"/>
0079	<Operation type="Enumeration" value="CreateKeyPair"/>

0080	<Operation type="Enumeration" value="ReKey"/>
0081	<Operation type="Enumeration" value="Archive"/>
0082	<Operation type="Enumeration" value="Recover"/>
0083	<Operation type="Enumeration" value="ObtainLease"/>
0084	<Operation type="Enumeration" value="Certify"/>
0085	<Operation type="Enumeration" value="ReCertify"/>
0086	<Operation type="Enumeration" value="Notify"/>
0087	<Operation type="Enumeration" value="Put"/>
0088	<ObjectType type="Enumeration" value="Certificate"/>
0089	<ObjectType type="Enumeration" value="SymmetricKey"/>
0090	<ObjectType type="Enumeration" value="SecretData"/>
0091	<ObjectType type="Enumeration" value="PublicKey"/>
0092	<ObjectType type="Enumeration" value="PrivateKey"/>
0093	<ObjectType type="Enumeration" value="Template"/>
0094	<ObjectType type="Enumeration" value="OpaqueObject"/>
0095	<ObjectType type="Enumeration" value="SplitKey"/>
0096	</ResponsePayload>
0097	</BatchItem>
0098	</ResponseMessage>
	<p>42007b01000002a042007a0100000048420069010000002042006a02000000040000000100000000</p> <p>42006b020000000400000000000000042009209000000080000000051caafbd42000d0200000004</p> <p>000000010000000042000f010000024842005c0500000004000000180000000042007f0500000004</p> <p>000000000000000042007c010000022042005c0500000004000000180000000042005c0500000004</p> <p>000000080000000042005c0500000004000000140000000042005c05000000040000000a00000000</p> <p>42005c0500000004000000010000000042005c0500000004000000030000000042005c0500000004</p> <p>0000000b0000000042005c05000000040000000c0000000042005c05000000040000000d00000000</p> <p>42005c05000000040000000e0000000042005c05000000040000000f0000000042005c0500000004</p> <p>000000120000000042005c0500000004000000130000000042005c05000000040000001a00000000</p> <p>42005c0500000004000000190000000042005c0500000004000000090000000042005c0500000004</p> <p>000000110000000042005c0500000004000000020000000042005c05000000040000000400000000</p> <p>42005c0500000004000000150000000042005c0500000004000000160000000042005c0500000004</p> <p>000000100000000042005c0500000004000000060000000042005c05000000040000000700000000</p> <p>42005c05000000040000001b0000000042005c05000000040000001c000000004200570500000004</p> <p>00000001000000004200570500000004000000020000000042005705000000040000000700000000</p> <p>42005705000000040000000300000000420057050000000400000004000000004200570500000004</p> <p>00000006000000004200570500000004000000080000000042005705000000040000000500000000</p>
	<pre> {"tag":"ResponseMessage", "value":{   {"tag":"ResponseHeader", "value":{     {"tag":"ProtocolVersion", "value":{       {"tag":"ProtocolVersionMajor", "type":"Integer", "value":"0x00000001"},       {"tag":"ProtocolVersionMinor", "type":"Integer", "value":"0x00000000"}     }},     {"tag":"TimeStamp", "type":"DateTime", "value":"2013-06-26T09:09:17+00:00"},     {"tag":"BatchCount", "type":"Integer", "value":"0x00000001"}   }},   {"tag":"BatchItem", "value":{     {"tag":"Operation", "type":"Enumeration", "value":"Query"},     {"tag":"ResultStatus", "type":"Enumeration", "value":"Success"},     {"tag":"ResponsePayload", "value":{       {"tag":"Operation", "type":"Enumeration", "value":"Query"},       {"tag":"Operation", "type":"Enumeration", "value":"Locate"},       {"tag":"Operation", "type":"Enumeration", "value":"Destroy"},       {"tag":"Operation", "type":"Enumeration", "value":"Get"},       {"tag":"Operation", "type":"Enumeration", "value":"Create"},       {"tag":"Operation", "type":"Enumeration", "value":"Register"},       {"tag":"Operation", "type":"Enumeration", "value":"GetAttributes"},       {"tag":"Operation", "type":"Enumeration", "value":"GetAttributeList"},       {"tag":"Operation", "type":"Enumeration", "value":"AddAttribute"},       {"tag":"Operation", "type":"Enumeration", "value":"ModifyAttribute"},       {"tag":"Operation", "type":"Enumeration", "value":"DeleteAttribute"},       {"tag":"Operation", "type":"Enumeration", "value":"Activate"},       {"tag":"Operation", "type":"Enumeration", "value":"Revoke"},       {"tag":"Operation", "type":"Enumeration", "value":"Poll"},       {"tag":"Operation", "type":"Enumeration", "value":"Cancel"},       {"tag":"Operation", "type":"Enumeration", "value":"Check"},     }}   } } </pre>

	<pre> {"tag": "Operation", "type": "Enumeration", "value": "GetUsageAllocation"}, {"tag": "Operation", "type": "Enumeration", "value": "CreateKeyPair"}, {"tag": "Operation", "type": "Enumeration", "value": "ReKey"}, {"tag": "Operation", "type": "Enumeration", "value": "Archive"}, {"tag": "Operation", "type": "Enumeration", "value": "Recover"}, {"tag": "Operation", "type": "Enumeration", "value": "ObtainLease"}, {"tag": "Operation", "type": "Enumeration", "value": "Certify"}, {"tag": "Operation", "type": "Enumeration", "value": "ReCertify"}, {"tag": "Operation", "type": "Enumeration", "value": "Notify"}, {"tag": "Operation", "type": "Enumeration", "value": "Put"}, {"tag": "ObjectType", "type": "Enumeration", "value": "Certificate"}, {"tag": "ObjectType", "type": "Enumeration", "value": "SymmetricKey"}, {"tag": "ObjectType", "type": "Enumeration", "value": "SecretData"}, {"tag": "ObjectType", "type": "Enumeration", "value": "PublicKey"}, {"tag": "ObjectType", "type": "Enumeration", "value": "PrivateKey"}, {"tag": "ObjectType", "type": "Enumeration", "value": "Template"}, {"tag": "ObjectType", "type": "Enumeration", "value": "OpaqueObject"}, {"tag": "ObjectType", "type": "Enumeration", "value": "SplitKey"}     ]}   ]} }</pre>
--	--

327

## 328 5.2 Mandatory JSON Profile Test Cases KMIP v1.1

### 329 5.2.1 MSGENC-JSON-M-1-11 - Query, Maximum Response Size

330 Perform a Query operation, querying the Operations and Objects supported by the server, with a  
 331 restriction on the Maximum Response Size set in the request header. Since the resulting Query response  
 332 is too big, an error is returned. Increase the Maximum Response Size, resubmit the Query request, and  
 333 get a successful response.

334 The specific list of operations and object types returned in the response MAY vary.

	<pre> # TIME 0 0001 &lt;RequestMessage&gt; 0002   &lt;RequestHeader&gt; 0003     &lt;ProtocolVersion&gt; 0004       &lt;ProtocolVersionMajor type="Integer" value="1"/&gt; 0005       &lt;ProtocolVersionMinor type="Integer" value="1"/&gt; 0006     &lt;/ProtocolVersion&gt; 0007     &lt;MaximumResponseSize type="Integer" value="256"/&gt; 0008     &lt;BatchCount type="Integer" value="1"/&gt; 0009   &lt;/RequestHeader&gt; 0010   &lt;BatchItem&gt; 0011     &lt;Operation type="Enumeration" value="Query"/&gt; 0012     &lt;RequestPayload&gt; 0013       &lt;QueryFunction type="Enumeration" value="QueryOperations"/&gt; 0014       &lt;QueryFunction type="Enumeration" value="QueryObjects"/&gt; 0015     &lt;/RequestPayload&gt; 0016   &lt;/BatchItem&gt; 0017 &lt;/RequestMessage&gt;  42007801000000904200770100000048420069010000002042006a02000000040000000100000000 42006b020000000400000001000000004200500200000004000001000000000042000d0200000004 000000010000000042000f010000003842005c050000000400000018000000004200790100000020 4200740500000004000000010000000042007405000000040000000200000000  {"tag": "RequestMessage", "value": [   {"tag": "RequestHeader", "value": [     {"tag": "ProtocolVersion", "value": [       {"tag": "ProtocolVersionMajor", "type": "Integer", "value": "0x00000001"},       {"tag": "ProtocolVersionMinor", "type": "Integer", "value": "0x00000001"}     ]}   ]} ]}</pre>
--	---



	<pre>     }},     {"tag":"MaximumResponseSize", "type":"Integer", "value":"0x00000100"},     {"tag":"BatchCount", "type":"Integer", "value":"0x00000001"}   ]},   {"tag":"BatchItem", "value":[     {"tag":"Operation", "type":"Enumeration", "value":"Query"},     {"tag":"RequestPayload", "value":[       {"tag":"QueryFunction", "type":"Enumeration", "value":"QueryOperations"},       {"tag":"QueryFunction", "type":"Enumeration", "value":"QueryObjects"}     ]}   ]} ]} </pre>
0018 0019 0020 0021 0022 0023 0024 0025 0026 0027 0028 0029 0030 0031 0032	<pre> &lt;ResponseMessage&gt;   &lt;ResponseHeader&gt;     &lt;ProtocolVersion&gt;       &lt;ProtocolVersionMajor type="Integer" value="1"/&gt;       &lt;ProtocolVersionMinor type="Integer" value="1"/&gt;     &lt;/ProtocolVersion&gt;     &lt;TimeStamp type="DateTime" value="2014-06-10T08:03:34+00:00"/&gt;     &lt;BatchCount type="Integer" value="1"/&gt;   &lt;/ResponseHeader&gt; &lt;BatchItem&gt;     &lt;Operation type="Enumeration" value="Query"/&gt;     &lt;ResultStatus type="Enumeration" value="OperationFailed"/&gt;     &lt;ResultReason type="Enumeration" value="ResponseTooLarge"/&gt;     &lt;ResultMessage type="TextString" value="TOO_LARGE"/&gt;   &lt;/BatchItem&gt; &lt;/ResponseMessage&gt; </pre> <p>42007b01000000a042007a0100000048420069010000002042006a02000000040000000100000000 42006b020000000400000001000000004200920900000008000000005396bc2442000d0200000004 000000010000000042000f010000004842005c0500000004000000180000000042007f0500000004 000000010000000042007e0500000004000000020000000042007d0700000009544f4f5f4c415247 4500000000000000</p> <pre> {"tag":"ResponseMessage", "value":[   {"tag":"ResponseHeader", "value":[     {"tag":"ProtocolVersion", "value":[       {"tag":"ProtocolVersionMajor", "type":"Integer", "value":"0x00000001"},       {"tag":"ProtocolVersionMinor", "type":"Integer", "value":"0x00000001"}     ]},     {"tag":"TimeStamp", "type":"DateTime", "value":"2014-06-10T08:04:52+00:00"},     {"tag":"BatchCount", "type":"Integer", "value":"0x00000001"}   ]},   {"tag":"BatchItem", "value":[     {"tag":"Operation", "type":"Enumeration", "value":"Query"},     {"tag":"ResultStatus", "type":"Enumeration", "value":"OperationFailed"},     {"tag":"ResultReason", "type":"Enumeration", "value":"ResponseTooLarge"},     {"tag":"ResultMessage", "type":"TextString", "value":"TOO_LARGE"}   ]} ]} </pre>
0033 0034 0035 0036 0037 0038 0039 0040 0041 0042 0043 0044	<pre> # TIME 1 &lt;RequestMessage&gt;   &lt;RequestHeader&gt;     &lt;ProtocolVersion&gt;       &lt;ProtocolVersionMajor type="Integer" value="1"/&gt;       &lt;ProtocolVersionMinor type="Integer" value="1"/&gt;     &lt;/ProtocolVersion&gt;     &lt;MaximumResponseSize type="Integer" value="2048"/&gt;     &lt;BatchCount type="Integer" value="1"/&gt;   &lt;/RequestHeader&gt;   &lt;BatchItem&gt;     &lt;Operation type="Enumeration" value="Query"/&gt;   &lt;RequestPayload&gt; </pre>



0045	<QueryFunction type="Enumeration" value="QueryOperations"/>
0046	<QueryFunction type="Enumeration" value="QueryObjects"/>
0047	</RequestPayload>
0048	</BatchItem>
0049	</RequestMessage>
	<p>42007801000000904200770100000048420069010000002042006a02000000040000000100000000</p> <p>42006b020000000400000001000000004200500200000004000008000000000042000d0200000004</p> <p>000000010000000042000f010000003842005c050000000400000018000000004200790100000020</p> <p>4200740500000004000000010000000042007405000000040000000200000000</p> <pre> {"tag": "RequestMessage", "value": [   {"tag": "RequestHeader", "value": [     {"tag": "ProtocolVersion", "value": [       {"tag": "ProtocolVersionMajor", "type": "Integer", "value": "0x00000001"},       {"tag": "ProtocolVersionMinor", "type": "Integer", "value": "0x00000001"}     ]},     {"tag": "MaximumResponseSize", "type": "Integer", "value": "0x00000800"},     {"tag": "BatchCount", "type": "Integer", "value": "0x00000001"}   ]},   {"tag": "BatchItem", "value": [     {"tag": "Operation", "type": "Enumeration", "value": "Query"},     {"tag": "RequestPayload", "value": [       {"tag": "QueryFunction", "type": "Enumeration", "value": "QueryOperations"},       {"tag": "QueryFunction", "type": "Enumeration", "value": "QueryObjects"}     ]}   ]} ]}</pre>
0050	<ResponseMessage>
0051	<ResponseHeader>
0052	<ProtocolVersion>
0053	<ProtocolVersionMajor type="Integer" value="1"/>
0054	<ProtocolVersionMinor type="Integer" value="1"/>
0055	</ProtocolVersion>
0056	<TimeStamp type="DateTime" value="2014-06-10T08:03:34+00:00"/>
0057	<BatchCount type="Integer" value="1"/>
0058	</ResponseHeader>
0059	<BatchItem>
0060	<Operation type="Enumeration" value="Query"/>
0061	<ResultStatus type="Enumeration" value="Success"/>
0062	<ResponsePayload>
0063	<Operation type="Enumeration" value="Query"/>
0064	<Operation type="Enumeration" value="Locate"/>
0065	<Operation type="Enumeration" value="Destroy"/>
0066	<Operation type="Enumeration" value="Get"/>
0067	<Operation type="Enumeration" value="Create"/>
0068	<Operation type="Enumeration" value="Register"/>
0069	<Operation type="Enumeration" value="GetAttributes"/>
0070	<Operation type="Enumeration" value="GetAttributeList"/>
0071	<Operation type="Enumeration" value="AddAttribute"/>
0072	<Operation type="Enumeration" value="ModifyAttribute"/>
0073	<Operation type="Enumeration" value="DeleteAttribute"/>
0074	<Operation type="Enumeration" value="Activate"/>
0075	<Operation type="Enumeration" value="Revoke"/>
0076	<Operation type="Enumeration" value="Poll"/>
0077	<Operation type="Enumeration" value="Cancel"/>
0078	<Operation type="Enumeration" value="Check"/>
0079	<Operation type="Enumeration" value="GetUsageAllocation"/>
0080	<Operation type="Enumeration" value="CreateKeyPair"/>
0081	<Operation type="Enumeration" value="ReKey"/>
0082	<Operation type="Enumeration" value="Archive"/>

0083	<Operation type="Enumeration" value="Recover"/>
0084	<Operation type="Enumeration" value="ObtainLease"/>
0085	<Operation type="Enumeration" value="ReKeyKeyPair"/>
0086	<Operation type="Enumeration" value="Certify"/>
0087	<Operation type="Enumeration" value="ReCertify"/>
0088	<Operation type="Enumeration" value="DiscoverVersions"/>
0089	<Operation type="Enumeration" value="Notify"/>
0090	<Operation type="Enumeration" value="Put"/>
0091	<ObjectType type="Enumeration" value="Certificate"/>
0092	<ObjectType type="Enumeration" value="SymmetricKey"/>
0093	<ObjectType type="Enumeration" value="SecretData"/>
0094	<ObjectType type="Enumeration" value="PublicKey"/>
0095	<ObjectType type="Enumeration" value="PrivateKey"/>
0096	<ObjectType type="Enumeration" value="Template"/>
0097	<ObjectType type="Enumeration" value="OpaqueObject"/>
0098	<ObjectType type="Enumeration" value="SplitKey"/>
0099	</ResponsePayload>
0100	</BatchItem>
0101	</ResponseMessage>
	<p>42007b01000002c042007a0100000048420069010000002042006a02000000040000000100000000</p> <p>42006b020000000400000001000000004200920900000008000000005396bc2442000d0200000004</p> <p>000000010000000042000f010000026842005c0500000004000000180000000042007f0500000004</p> <p>000000000000000042007c010000024042005c0500000004000000180000000042005c0500000004</p> <p>000000080000000042005c0500000004000000140000000042005c05000000040000000a00000000</p> <p>42005c0500000004000000010000000042005c0500000004000000030000000042005c0500000004</p> <p>0000000b0000000042005c05000000040000000c0000000042005c05000000040000000d00000000</p> <p>42005c05000000040000000e0000000042005c05000000040000000f0000000042005c0500000004</p> <p>000000120000000042005c0500000004000000130000000042005c05000000040000001a00000000</p> <p>42005c0500000004000000190000000042005c0500000004000000090000000042005c0500000004</p> <p>000000110000000042005c0500000004000000020000000042005c05000000040000000400000000</p> <p>42005c0500000004000000150000000042005c0500000004000000160000000042005c0500000004</p> <p>000000100000000042005c05000000040000001d0000000042005c05000000040000000600000000</p> <p>42005c0500000004000000070000000042005c05000000040000001e0000000042005c0500000004</p> <p>0000001b0000000042005c05000000040000001c0000000042005705000000040000000100000000</p> <p>42005705000000040000000200000000420057050000000400000007000000004200570500000004</p> <p>00000003000000004200570500000004000000040000000042005705000000040000000600000000</p> <p>4200570500000004000000080000000042005705000000040000000500000000</p> <pre> {"tag":"ResponseMessage", "value":{   {"tag":"ResponseHeader", "value":{     {"tag":"ProtocolVersion", "value":{       {"tag":"ProtocolVersionMajor", "type":"Integer", "value":"0x00000001"},       {"tag":"ProtocolVersionMinor", "type":"Integer", "value":"0x00000001"}     }},     {"tag":"TimeStamp", "type":"DateTime", "value":"2014-06-10T08:04:52+00:00"},     {"tag":"BatchCount", "type":"Integer", "value":"0x00000001"}   }},   {"tag":"BatchItem", "value":{     {"tag":"Operation", "type":"Enumeration", "value":"Query"},     {"tag":"ResultStatus", "type":"Enumeration", "value":"Success"},     {"tag":"ResponsePayload", "value":{       {"tag":"Operation", "type":"Enumeration", "value":"Query"},       {"tag":"Operation", "type":"Enumeration", "value":"Locate"},       {"tag":"Operation", "type":"Enumeration", "value":"Destroy"},       {"tag":"Operation", "type":"Enumeration", "value":"Get"},       {"tag":"Operation", "type":"Enumeration", "value":"Create"},       {"tag":"Operation", "type":"Enumeration", "value":"Register"},       {"tag":"Operation", "type":"Enumeration", "value":"GetAttributes"},       {"tag":"Operation", "type":"Enumeration", "value":"GetAttributeList"},       {"tag":"Operation", "type":"Enumeration", "value":"AddAttribute"},       {"tag":"Operation", "type":"Enumeration", "value":"ModifyAttribute"},       {"tag":"Operation", "type":"Enumeration", "value":"DeleteAttribute"},       {"tag":"Operation", "type":"Enumeration", "value":"Activate"},       {"tag":"Operation", "type":"Enumeration", "value":"Revoke"},       {"tag":"Operation", "type":"Enumeration", "value":"Poll"},       {"tag":"Operation", "type":"Enumeration", "value":"Cancel"},     }}   }} } </pre>

	<pre> {"tag": "Operation", "type": "Enumeration", "value": "Check"}, {"tag": "Operation", "type": "Enumeration", "value": "GetUsageAllocation"}, {"tag": "Operation", "type": "Enumeration", "value": "CreateKeyPair"}, {"tag": "Operation", "type": "Enumeration", "value": "ReKey"}, {"tag": "Operation", "type": "Enumeration", "value": "Archive"}, {"tag": "Operation", "type": "Enumeration", "value": "Recover"}, {"tag": "Operation", "type": "Enumeration", "value": "ObtainLease"}, {"tag": "Operation", "type": "Enumeration", "value": "ReKeyKeyPair"}, {"tag": "Operation", "type": "Enumeration", "value": "Certify"}, {"tag": "Operation", "type": "Enumeration", "value": "ReCertify"}, {"tag": "Operation", "type": "Enumeration", "value": "DiscoverVersions"}, {"tag": "Operation", "type": "Enumeration", "value": "Notify"}, {"tag": "Operation", "type": "Enumeration", "value": "Put"}, {"tag": "ObjectType", "type": "Enumeration", "value": "Certificate"}, {"tag": "ObjectType", "type": "Enumeration", "value": "SymmetricKey"}, {"tag": "ObjectType", "type": "Enumeration", "value": "SecretData"}, {"tag": "ObjectType", "type": "Enumeration", "value": "PublicKey"}, {"tag": "ObjectType", "type": "Enumeration", "value": "PrivateKey"}, {"tag": "ObjectType", "type": "Enumeration", "value": "Template"}, {"tag": "ObjectType", "type": "Enumeration", "value": "OpaqueObject"}, {"tag": "ObjectType", "type": "Enumeration", "value": "SplitKey"}     ]}   ]} }] </pre>
--	---

335

## 336 5.3 Mandatory JSON Profile Test Cases KMIP v1.2

### 337 5.3.1 MSGENC-JSON-M-1-12 - Query, Maximum Response Size

338 Perform a Query operation, querying the Operations and Objects supported by the server, with a  
 339 restriction on the Maximum Response Size set in the request header. Since the resulting Query response  
 340 is too big, an error is returned. Increase the Maximum Response Size, resubmit the Query request, and  
 341 get a successful response.

342 The specific list of operations and object types returned in the response MAY vary.

	<pre> # TIME 0 0001 &lt;RequestMessage&gt; 0002   &lt;RequestHeader&gt; 0003     &lt;ProtocolVersion&gt; 0004       &lt;ProtocolVersionMajor type="Integer" value="1"/&gt; 0005       &lt;ProtocolVersionMinor type="Integer" value="2"/&gt; 0006     &lt;/ProtocolVersion&gt; 0007     &lt;MaximumResponseSize type="Integer" value="256"/&gt; 0008     &lt;BatchCount type="Integer" value="1"/&gt; 0009   &lt;/RequestHeader&gt; 0010   &lt;BatchItem&gt; 0011     &lt;Operation type="Enumeration" value="Query"/&gt; 0012     &lt;RequestPayload&gt; 0013       &lt;QueryFunction type="Enumeration" value="QueryOperations"/&gt; 0014       &lt;QueryFunction type="Enumeration" value="QueryObjects"/&gt; 0015     &lt;/RequestPayload&gt; 0016   &lt;/BatchItem&gt; 0017 &lt;/RequestMessage&gt;  420078010000000904200770100000048420069010000002042006a02000000040000000100000000 42006b020000000400000002000000004200500200000004000001000000000042000d0200000004 000000010000000042000f010000003842005c050000000400000018000000004200790100000020 4200740500000004000000010000000042007405000000040000000200000000  {"tag": "RequestMessage", "value": [   {"tag": "RequestHeader", "value": [     {"tag": "ProtocolVersion", "value": [ </pre>
--	---

	<pre> {"tag":"ProtocolVersionMajor", "type":"Integer", "value":"0x00000001"}, {"tag":"ProtocolVersionMinor", "type":"Integer", "value":"0x00000002"} ]], {"tag":"MaximumResponseSize", "type":"Integer", "value":"0x00000100"}, {"tag":"BatchCount", "type":"Integer", "value":"0x00000001"} ]], {"tag":"BatchItem", "value":[ {"tag":"Operation", "type":"Enumeration", "value":"Query"}, {"tag":"RequestPayload", "value":[ {"tag":"QueryFunction", "type":"Enumeration", "value":"QueryOperations"}, {"tag":"QueryFunction", "type":"Enumeration", "value":"QueryObjects"} ]] ]] ]] </pre>
0018 0019 0020 0021 0022 0023 0024 0025 0026 0027 0028 0029 0030 0031 0032	<pre> &lt;ResponseMessage&gt;   &lt;ResponseHeader&gt;     &lt;ProtocolVersion&gt;       &lt;ProtocolVersionMajor type="Integer" value="1"/&gt;       &lt;ProtocolVersionMinor type="Integer" value="2"/&gt;     &lt;/ProtocolVersion&gt;     &lt;TimeStamp type="DateTime" value="2014-06-10T08:07:28+00:00"/&gt;     &lt;BatchCount type="Integer" value="1"/&gt;   &lt;/ResponseHeader&gt; &lt;BatchItem&gt;     &lt;Operation type="Enumeration" value="Query"/&gt;     &lt;ResultStatus type="Enumeration" value="OperationFailed"/&gt;     &lt;ResultReason type="Enumeration" value="ResponseTooLarge"/&gt;     &lt;ResultMessage type="TextString" value="TOO_LARGE"/&gt;   &lt;/BatchItem&gt; &lt;/ResponseMessage&gt; </pre> <p>42007b01000000a042007a0100000048420069010000002042006a02000000040000000100000000 42006b0200000004000000002000000004200920900000008000000005396bcc042000d0200000004 0000000100000000042000f010000004842005c0500000004000000180000000042007f0500000004 0000000100000000042007e0500000004000000020000000042007d0700000009544f4f5f4c415247 4500000000000000</p> <pre> {"tag":"ResponseMessage", "value":[ {"tag":"ResponseHeader", "value":[ {"tag":"ProtocolVersion", "value":[ {"tag":"ProtocolVersionMajor", "type":"Integer", "value":"0x00000001"}, {"tag":"ProtocolVersionMinor", "type":"Integer", "value":"0x00000002"} ]], {"tag":"TimeStamp", "type":"DateTime", "value":"2014-06-10T08:07:28+00:00"}, {"tag":"BatchCount", "type":"Integer", "value":"0x00000001"} ]], {"tag":"BatchItem", "value":[ {"tag":"Operation", "type":"Enumeration", "value":"Query"}, {"tag":"ResultStatus", "type":"Enumeration", "value":"OperationFailed"}, {"tag":"ResultReason", "type":"Enumeration", "value":"ResponseTooLarge"}, {"tag":"ResultMessage", "type":"TextString", "value":"TOO_LARGE"} ]] ]] </pre>
0033 0034 0035 0036 0037 0038 0039 0040 0041 0042	<pre> # TIME 1 &lt;RequestMessage&gt;   &lt;RequestHeader&gt;     &lt;ProtocolVersion&gt;       &lt;ProtocolVersionMajor type="Integer" value="1"/&gt;       &lt;ProtocolVersionMinor type="Integer" value="2"/&gt;     &lt;/ProtocolVersion&gt;     &lt;MaximumResponseSize type="Integer" value="2048"/&gt;     &lt;BatchCount type="Integer" value="1"/&gt;   &lt;/RequestHeader&gt;   &lt;BatchItem&gt; </pre>

0043	<Operation type="Enumeration" value="Query"/>
0044	<RequestPayload>
0045	<QueryFunction type="Enumeration" value="QueryOperations"/>
0046	<QueryFunction type="Enumeration" value="QueryObjects"/>
0047	</RequestPayload>
0048	</BatchItem>
0049	</RequestMessage>
	<p>42007801000000904200770100000048420069010000002042006a02000000040000000100000000  42006b020000000400000002000000004200500200000004000008000000000042000d0200000004  000000010000000042000f010000003842005c050000000400000018000000004200790100000020  4200740500000004000000010000000042007405000000040000000200000000</p> <pre>{   "tag": "RequestMessage",   "value": [     {       "tag": "RequestHeader",       "value": [         {           "tag": "ProtocolVersion",           "value": [             {               "tag": "ProtocolVersionMajor",               "type": "Integer",               "value": "0x00000001"             },             {               "tag": "ProtocolVersionMinor",               "type": "Integer",               "value": "0x00000002"             }           ]         },         {           "tag": "MaximumResponseSize",           "type": "Integer",           "value": "0x00000800"         },         {           "tag": "BatchCount",           "type": "Integer",           "value": "0x00000001"         }       ]     },     {       "tag": "BatchItem",       "value": [         {           "tag": "Operation",           "type": "Enumeration",           "value": "Query"         },         {           "tag": "RequestPayload",           "value": [             {               "tag": "QueryFunction",               "type": "Enumeration",               "value": "QueryOperations"             },             {               "tag": "QueryFunction",               "type": "Enumeration",               "value": "QueryObjects"             }           ]         }       ]     }   ] }</pre>
0050	<ResponseMessage>
0051	<ResponseHeader>
0052	<ProtocolVersion>
0053	<ProtocolVersionMajor type="Integer" value="1"/>
0054	<ProtocolVersionMinor type="Integer" value="2"/>
0055	</ProtocolVersion>
0056	<TimeStamp type="DateTime" value="2014-06-10T08:07:28+00:00"/>
0057	<BatchCount type="Integer" value="1"/>
0058	</ResponseHeader>
0059	<BatchItem>
0060	<Operation type="Enumeration" value="Query"/>
0061	<ResultStatus type="Enumeration" value="Success"/>
0062	<ResponsePayload>
0063	<Operation type="Enumeration" value="Query"/>
0064	<Operation type="Enumeration" value="Locate"/>
0065	<Operation type="Enumeration" value="Destroy"/>
0066	<Operation type="Enumeration" value="Get"/>
0067	<Operation type="Enumeration" value="Create"/>
0068	<Operation type="Enumeration" value="Register"/>
0069	<Operation type="Enumeration" value="GetAttributes"/>
0070	<Operation type="Enumeration" value="GetAttributeList"/>
0071	<Operation type="Enumeration" value="AddAttribute"/>
0072	<Operation type="Enumeration" value="ModifyAttribute"/>
0073	<Operation type="Enumeration" value="DeleteAttribute"/>
0074	<Operation type="Enumeration" value="Activate"/>
0075	<Operation type="Enumeration" value="Revoke"/>
0076	<Operation type="Enumeration" value="Poll"/>
0077	<Operation type="Enumeration" value="Cancel"/>
0078	<Operation type="Enumeration" value="Check"/>
0079	<Operation type="Enumeration" value="GetUsageAllocation"/>
0080	<Operation type="Enumeration" value="CreateKeyPair"/>

0081	<Operation type="Enumeration" value="ReKey"/>
0082	<Operation type="Enumeration" value="Archive"/>
0083	<Operation type="Enumeration" value="Recover"/>
0084	<Operation type="Enumeration" value="ObtainLease"/>
0085	<Operation type="Enumeration" value="ReKeyKeyPair"/>
0086	<Operation type="Enumeration" value="Certify"/>
0087	<Operation type="Enumeration" value="ReCertify"/>
0088	<Operation type="Enumeration" value="DiscoverVersions"/>
0089	<Operation type="Enumeration" value="Notify"/>
0090	<Operation type="Enumeration" value="Put"/>
0091	<Operation type="Enumeration" value="RNGRetrieve"/>
0092	<Operation type="Enumeration" value="RNGSeed"/>
0093	<Operation type="Enumeration" value="Encrypt"/>
0094	<Operation type="Enumeration" value="Decrypt"/>
0095	<Operation type="Enumeration" value="Sign"/>
0096	<Operation type="Enumeration" value="SignatureVerify"/>
0097	<Operation type="Enumeration" value="MAC"/>
0098	<Operation type="Enumeration" value="MACVerify"/>
0099	<Operation type="Enumeration" value="Hash"/>
0100	<Operation type="Enumeration" value="CreateSplitKey"/>
0101	<Operation type="Enumeration" value="JoinSplitKey"/>
0102	<ObjectType type="Enumeration" value="Certificate"/>
0103	<ObjectType type="Enumeration" value="SymmetricKey"/>
0104	<ObjectType type="Enumeration" value="SecretData"/>
0105	<ObjectType type="Enumeration" value="PublicKey"/>
0106	<ObjectType type="Enumeration" value="PrivateKey"/>
0107	<ObjectType type="Enumeration" value="Template"/>
0108	<ObjectType type="Enumeration" value="OpaqueObject"/>
0109	<ObjectType type="Enumeration" value="SplitKey"/>
0110	<ObjectType type="Enumeration" value="PGPKey"/>
0111	</ResponsePayload>
0112	</BatchItem>
0113	</ResponseMessage>
	42007b010000038042007a0100000048420069010000002042006a02000000040000000100000000 42006b0200000004000000002000000004200920900000008000000005396bcc042000d0200000004 000000010000000042000f010000032842005c0500000004000000180000000042007f0500000004 000000000000000042007c010000030042005c0500000004000000180000000042005c0500000004 000000080000000042005c0500000004000000140000000042005c05000000040000000a00000000 42005c0500000004000000010000000042005c0500000004000000030000000042005c0500000004 0000000b0000000042005c05000000040000000c0000000042005c05000000040000000d00000000 42005c05000000040000000e0000000042005c05000000040000000f0000000042005c0500000004 000000120000000042005c0500000004000000130000000042005c05000000040000001a00000000 42005c0500000004000000190000000042005c0500000004000000090000000042005c0500000004 000000110000000042005c0500000004000000020000000042005c05000000040000000400000000 42005c0500000004000000150000000042005c0500000004000000160000000042005c0500000004 000000100000000042005c05000000040000001d0000000042005c05000000040000000600000000 42005c0500000004000000070000000042005c05000000040000001e0000000042005c0500000004 0000001b0000000042005c05000000040000001c0000000042005c05000000040000000250000000 42005c05000000040000000260000000042005c05000000040000001f0000000042005c0500000004 000000200000000042005c0500000004000000210000000042005c05000000040000002200000000 42005c0500000004000000230000000042005c0500000004000000240000000042005c0500000004 000000270000000042005c0500000004000000280000000042005c05000000040000002900000000 42005c0500000004000000010000000042005705000000040000002000000004200570500000004 00000007000000004200570500000004000000030000000042005705000000040000000400000000 42005705000000040000000600000000420057050000000400000008000000004200570500000004 000000050000000042005705000000040000000900000000  {"tag":"ResponseMessage", "value": {"tag":"ResponseHeader", "value": {"tag":"ProtocolVersion", "value": {"tag":"ProtocolVersionMajor", "type":"Integer", "value":"0x00000001"}, {"tag":"ProtocolVersionMinor", "type":"Integer", "value":"0x00000002"} }, }, }

	<pre> {"tag":"TimeStamp", "type":"DateTime", "value":"2014-06-10T08:07:28+00:00"}, {"tag":"BatchCount", "type":"Integer", "value":"0x00000001"} }}, {"tag":"BatchItem", "value":[ {"tag":"Operation", "type":"Enumeration", "value":"Query"}, {"tag":"ResultStatus", "type":"Enumeration", "value":"Success"}, {"tag":"ResponsePayload", "value":[ {"tag":"Operation", "type":"Enumeration", "value":"Query"}, {"tag":"Operation", "type":"Enumeration", "value":"Locate"}, {"tag":"Operation", "type":"Enumeration", "value":"Destroy"}, {"tag":"Operation", "type":"Enumeration", "value":"Get"}, {"tag":"Operation", "type":"Enumeration", "value":"Create"}, {"tag":"Operation", "type":"Enumeration", "value":"Register"}, {"tag":"Operation", "type":"Enumeration", "value":"GetAttributes"}, {"tag":"Operation", "type":"Enumeration", "value":"GetAttributeList"}, {"tag":"Operation", "type":"Enumeration", "value":"AddAttribute"}, {"tag":"Operation", "type":"Enumeration", "value":"ModifyAttribute"}, {"tag":"Operation", "type":"Enumeration", "value":"DeleteAttribute"}, {"tag":"Operation", "type":"Enumeration", "value":"Activate"}, {"tag":"Operation", "type":"Enumeration", "value":"Revoke"}, {"tag":"Operation", "type":"Enumeration", "value":"Poll"}, {"tag":"Operation", "type":"Enumeration", "value":"Cancel"}, {"tag":"Operation", "type":"Enumeration", "value":"Check"}, {"tag":"Operation", "type":"Enumeration", "value":"GetUsageAllocation"}, {"tag":"Operation", "type":"Enumeration", "value":"CreateKeyPair"}, {"tag":"Operation", "type":"Enumeration", "value":"ReKey"}, {"tag":"Operation", "type":"Enumeration", "value":"Archive"}, {"tag":"Operation", "type":"Enumeration", "value":"Recover"}, {"tag":"Operation", "type":"Enumeration", "value":"ObtainLease"}, {"tag":"Operation", "type":"Enumeration", "value":"ReKeyKeyPair"}, {"tag":"Operation", "type":"Enumeration", "value":"Certify"}, {"tag":"Operation", "type":"Enumeration", "value":"ReCertify"}, {"tag":"Operation", "type":"Enumeration", "value":"DiscoverVersions"}, {"tag":"Operation", "type":"Enumeration", "value":"Notify"}, {"tag":"Operation", "type":"Enumeration", "value":"Put"}, {"tag":"Operation", "type":"Enumeration", "value":"RNGRetrieve"}, {"tag":"Operation", "type":"Enumeration", "value":"RNGSeed"}, {"tag":"Operation", "type":"Enumeration", "value":"Encrypt"}, {"tag":"Operation", "type":"Enumeration", "value":"Decrypt"}, {"tag":"Operation", "type":"Enumeration", "value":"Sign"}, {"tag":"Operation", "type":"Enumeration", "value":"SignatureVerify"}, {"tag":"Operation", "type":"Enumeration", "value":"MAC"}, {"tag":"Operation", "type":"Enumeration", "value":"MACVerify"}, {"tag":"Operation", "type":"Enumeration", "value":"Hash"}, {"tag":"Operation", "type":"Enumeration", "value":"CreateSplitKey"}, {"tag":"Operation", "type":"Enumeration", "value":"JoinSplitKey"}, {"tag":"ObjectType", "type":"Enumeration", "value":"Certificate"}, {"tag":"ObjectType", "type":"Enumeration", "value":"SymmetricKey"}, {"tag":"ObjectType", "type":"Enumeration", "value":"SecretData"}, {"tag":"ObjectType", "type":"Enumeration", "value":"PublicKey"}, {"tag":"ObjectType", "type":"Enumeration", "value":"PrivateKey"}, {"tag":"ObjectType", "type":"Enumeration", "value":"Template"}, {"tag":"ObjectType", "type":"Enumeration", "value":"OpaqueObject"}, {"tag":"ObjectType", "type":"Enumeration", "value":"SplitKey"}, {"tag":"ObjectType", "type":"Enumeration", "value":"PGPKey"} ]} ]} ]] </pre>
--	--



---

## 6 XML Profile

The XML profile specifies the use of KMIP replacing the TTLV message encoding with an XML message encoding. The results returned using the XML encoding SHALL be logically the same as if the message encoding was in TTLV form. All size or length values specified within tag values for KMIP items SHALL be the same in XML form as if the message encoding were in TTLV form. The implications of this are that items such as MaximumResponseSize are interpreted to refer to a maximum length computed as if it were a TTLV-encoded response, not the length of the JSON-encoded response.

### 6.1 XML Encoding

#### 6.1.1 Hex representations

Hex representations of numbers must always begin with '0x' and must not include any spaces. They may use either upper or lower case 'a'-'f'. The hex representation must include all leading zeros or sign extension bits when representing a value of a fixed width such as Tags (3 bytes), Integer (32-bit signed big-endian), Long Integer (64-bit signed big-endian) and Big Integer (big-endian multiple of 8 bytes). The Integer values for -1, 0, 1 are represented as "0xffffffff", "0x00000000", "0x00000001". Hex representation for Byte Strings are similar to numbers, but do not include the '0x' prefix, and can be of any length.

#### 6.1.2 Tags

Tags are a String that may contain either:

- The 3-byte tag hex value prefixed with '0x'
- The normalised text of a Tag as specified in the KMIP Specification

Other text values may be used such as published names of Extension tags, or names of new tags added in future KMIP versions. Producers may however choose to use hex values for these tags to ensure they are understood by all consumers.

#### 6.1.3 Normalizing Names

KMIP text values of Tags, Types and Enumerations SHALL be normalized to create a 'CamelCase' format that would be suitable to be used as a variable name in C/Java or an XML element name.

The basic approach to converting from KMIP text to CamelCase is to separate the text into individual word tokens (rules 1-4), capitalize the first letter of each word (rule 5) and then join with spaces removed (rule 6). The tokenizing splits on whitespace and on dashes where the token following is a valid word. The tokenizing also removes round brackets and shifts decimals from the front to the back of the first word in each string. The following rules SHALL be applied to create the normalized CamelCase form:

7. Replace round brackets '(', ')' with spaces
8. If a non-word char (not alpha, digit or underscore) is followed by a letter (either upper or lower case) then a lower case letter, replace the non-word char with space
9. Replace remaining non-word chars (except whitespace) with underscore.
10. If the first word begins with a digit, move all digits at start of first word to end of first word
11. Capitalize the first letter of each word
12. Concatenate all words with spaces removed



```

383 # 1. Replace brackets with space
384 noBrackets = re.sub('([()])', ' ', enumName)
385 # 2. replace \W with space if followed by letter, lower
386 nonWordToSpace = re.sub('\W([A-Za-z][a-z])', r' \1', noBrackets)
387 # 3. non-word to underscore
388 words = [re.sub('\W', '_', s) for s in nonWordToSpace.split()]
389 # 4. move numbers to end of first word
390 words[0] = re.sub('^(\\d+)(.*)', r'\\2\\1', words[0])
391 # 5. captialize first letter of each word
392 words = [re.sub('^.\\.', s[0].upper(), s) for s in words]
393 # 6. concatenate
394 enumNameCamel = ''.join(words)

```

*Example python name normalization code*

```

397 # 1. Replace brackets with space
398 $enumName=~s/[\\(\\)]/ /g;
399 # 2. replace \W with space if followed by letter, lower
400 $enumName=~s/\\W([A-Za-z][a-z])/ \\1/g;
401 # 3. non-word to underscore
402 @words=split(/ /,$enumName);
403 for($i=0;$i<=$#words;$i++) { $words[$i]=~s/\\W/_/g; }
404 # 4. move numbers to end of first word
405 $words[0] =~ s/^(\\d+)(.*)/\\2\\1/;
406 # 5. captialize first letter of each word
407 for($i=0;$i<=$#words;$i++) {
408     substr($words[$i],0,1)=~tr/a-z/A-Z/;
409 }
410 # 6. concatenate
411 $enumNameCamel = join(' ',@words);
412

```

*Example perl name normalization code*

## 6.1.4 Type

Type must be a String containing one of the normalized CamelCase values as defined in the KMIP specification.

- Structure
- Integer
- LongInteger
- BigInteger
- Enumeration
- Boolean
- TextString
- ByteString
- DateTime
- Interval

If type is not included, the default type of Structure SHALL be used.

## 6.1.5 Value

The specification of a value is represented differently for each TTLV type.

## 6.1.6 XML Element Encoding

For XML, each TTLV is represented as an XML element with attributes. The general form uses a single element named 'TTLV' with 'tag', optional 'name' and 'type' attributes. This form allows any TTLV including extensions to be encoded. For tags defined in the KMIP Specification or other well-known extensions, a more specific form can be used where each tag is encoded as an element with the same name and includes a 'type' attribute. For either form, structure values are encoded as nested xml elements, and non-structure values are encoded using the 'value' attribute.

```
<TTLV tag="0x420001" name="ActivationDate" type="DateTime" value="2001-01-01T10:00:00+10:00"/>
<TTLV tag="0x420001" type="DateTime" value="2001-01-01T10:00:00+10:00"/>
<ActivationDate type="DateTime" value="2001-01-01T10:00:00+10:00"/>
<TTLV tag="0x54FFFF" name="SomeExtension" type="DateTime" value="2001-01-01T10:00:00+10:00"/>
```

The 'type' property / attribute SHALL have a default value of 'Structure' and may be omitted for Structures.

If namespaces are required, XML elements SHALL use the following namespace:

urn:oasis:tc:kmip:xmlns

### 6.1.6.1 Tags

Tags are a String that may contain either:

- The 3-byte tag hex value prefixed with '0x'
- The normalised text of a Tag as specified in the KMIP Specification

Other text values may be used such as published names of Extension tags, or names of new tags added in future KMIP versions. Producers may however choose to use hex values for these tags to ensure they are understood by all consumers.

```
<ActivationDate xmlns="urn:oasis:tc:kmip:xmlns" type="DateTime" value="2001-01-01T10:00:00+10:00"/>
<IVCounterNonce type="ByteString" value="alb2c3d4"/>
<PrivateKeyTemplateAttribute type="Structure"/>
<TTLV tag="0x545352" name="SomeExtension" type="TextString" value="This is an extension"/>
<WELL_KNOWN_EXTENSION type="TextString" value="This is an extension"/>
```

### 6.1.6.2 Structure

For XML, sub-items are nested elements.

```
<ProtocolVersion type="Structure">
  <ProtocolVersionMajor type="Integer" value="1"/>
  <ProtocolVersionMinor type="Integer" value="0"/>
</ProtocolVersion>
<ProtocolVersion>
  <ProtocolVersionMajor type="Integer" value="1"/>
  <ProtocolVersionMinor type="Integer" value="0"/>
</ProtocolVersion>
```

The 'type' property / attribute is optional for a Structure.

### 6.1.6.3 Integer

For XML, value is a decimal and uses [XML-SCHEMA] type xsd:int

```
<BatchCount type="Integer" value="10"/>
```

### 6.1.6.4 Integer - Special case for Masks

(Cryptographic Usage Mask, Storage Status Mask):

Integer mask values can also be encoded as a String containing mask components. XML uses an attribute with [XML-SCHEMA] type xsd:list which uses a space separator. Components may be either the text of the enumeration value as defined in [KMIP 9.1.3.3.1](#) / [KMIP 9.1.3.3.2](#), or a 32-bit unsigned big-endian hex string.

```
<CryptographicUsageMask type="Integer" value="0x0000100c"/>
<CryptographicUsageMask type="Integer" value="Encrypt Decrypt CertificateSign"/>
<CryptographicUsageMask type="Integer" value="CertificateSign 0x00000004 0x00000008"/>
<CryptographicUsageMask type="Integer" value="CertificateSign 0x0000000c"/>
```

### 6.1.6.5 Long Integer

For XML, value uses [XML-SCHEMA] type xsd:long

```
<x540001 type="LongInteger" value="-2"/>
<UsageLimitsCount type="LongInteger" value="1152921504606846976"/>
```

### 6.1.6.6 Big Integer

For XML, value uses [XML-SCHEMA] type xsd:hexBinary

```
<X type="BigInteger" value="0000000000000000"/>
```

### 6.1.6.7 Enumeration

For XML, value uses [XML-SCHEMA] type xsd:string and is either a hex string or the CamelCase enum text. If an XSD with xsd:enumeration restriction is used to define valid values (as is the case with the XSD included as an appendix), parsers should also accept any hex string in addition to defined enum values.

```
<ObjectType type="Enumeration" value="0x00000002"/>
<ObjectType type="Enumeration" value="SymmetricKey"/>
```

### 6.1.6.8 Boolean

For XML, value uses [XML-SCHEMA] type xsd:Boolean

```
<BatchOrderOption type="Boolean" value="true"/>
```

### 6.1.6.9 Text String

XML uses [XML-SCHEMA] type xsd:string

```
<AttributeName type="TextString" value="Cryptographic Algorithm"/>
```

### 6.1.6.10 Byte String

XML uses [XML-SCHEMA] type xsd:hexBinary

```
<MACSignature type="ByteString" value="C50F77"/>
```

### 6.1.6.11 Date-Time

For XML, value uses [XML-SCHEMA] type xsd:dateTime

```
<ArchiveDate type="DateTime" value="2001-01-01T10:00:00+10:00"/>
```

### 6.1.6.12 Interval

XML uses [XML-SCHEMA] type xsd:unsignedInt

```
<Offset type="Interval" value="27"/>
```

## 7 XML Profile Test Cases

The test cases define a number of request-response pairs for KMIP operations. Each test case is provided in the XML format specified in this section intended to be both human-readable and usable by automated tools. The time sequence (starting from 0) for each request-response pair is noted and line numbers are provided for ease of cross-reference for a given test sequence.

Each test case has a unique label (the section name) which includes indication of mandatory (-M-) or optional (-O-) status and the protocol version major and minor numbers as part of the identifier.

The test cases may depend on a specific configuration of a KMIP client and server being configured in a manner consistent with the test case assumptions.

Where possible the flow of unique identifiers between tests, the date-time values, and other dynamic items are indicated using symbolic identifiers – in actual request and response messages these dynamic values will be filled in with valid values.

Note: the values for the returned items and the custom attributes are illustrative. Actual values from a real client system may vary as specified in section 8.4

### 7.1 Mandatory XML Profile Test Cases KMIP v1.0

#### 7.1.1 MSGENC-XML-M-1-10 - Query, Maximum Response Size

Perform a Query operation, querying the Operations and Objects supported by the server, with a restriction on the Maximum Response Size set in the request header. Since the resulting Query response is too big, an error is returned. Increase the Maximum Response Size, resubmit the Query request, and get a successful response.

The specific list of operations and object types returned in the response MAY vary.

0001	# TIME 0
0002	<RequestMessage>
0003	<RequestHeader>
0004	<ProtocolVersion>
0005	<ProtocolVersionMajor type="Integer" value="1"/>
0006	<ProtocolVersionMinor type="Integer" value="0"/>
0007	</ProtocolVersion>
0008	<MaximumResponseSize type="Integer" value="256"/>
0009	<BatchCount type="Integer" value="1"/>
0010	</RequestHeader>
0011	<BatchItem>
0012	<Operation type="Enumeration" value="Query"/>
0013	<RequestPayload>
0014	<QueryFunction type="Enumeration" value="QueryOperations"/>
0015	<QueryFunction type="Enumeration" value="QueryObjects"/>
0016	</RequestPayload>
0017	</BatchItem>
0018	</RequestMessage>
42007801000000904200770100000048420069010000002042006a02000000040000000100000000 42006b0200000004000000000000000042005002000000040000001000000000042000d0200000004 000000010000000042000f010000003842005c050000000400000018000000004200790100000020 4200740500000004000000010000000042007405000000040000000200000000	
0018	<ResponseMessage>
0019	<ResponseHeader>
0020	<ProtocolVersion>
0021	<ProtocolVersionMajor type="Integer" value="1"/>

0022	<ProtocolVersionMinor type="Integer" value="0"/>
0023	</ProtocolVersion>
0024	<TimeStamp type="DateTime" value="2013-06-26T09:09:17+00:00"/>
0025	<BatchCount type="Integer" value="1"/>
0026	</ResponseHeader> <BatchItem>
0027	<Operation type="Enumeration" value="Query"/>
0028	<ResultStatus type="Enumeration" value="OperationFailed"/>
0029	<ResultReason type="Enumeration" value="ResponseTooLarge"/>
0030	<ResultMessage type="TextString" value="TOO_LARGE"/>
0031	</BatchItem>
0032	</ResponseMessage>
	42007b01000000a042007a0100000048420069010000002042006a02000000040000000100000000 42006b0200000004000000000000000042009209000000080000000051caafbd42000d0200000004 000000010000000042000f010000004842005c0500000004000000180000000042007f0500000004 000000010000000042007e0500000004000000020000000042007d0700000009544f4f5f4c415247 4500000000000000
0032	# TIME 1 <RequestMessage>
0033	<RequestHeader>
0034	<ProtocolVersion>
0035	<ProtocolVersionMajor type="Integer" value="1"/>
0036	<ProtocolVersionMinor type="Integer" value="0"/>
0037	</ProtocolVersion>
0038	<MaximumResponseSize type="Integer" value="2048"/>
0039	<BatchCount type="Integer" value="1"/>
0040	</RequestHeader>
0041	<BatchItem>
0042	<Operation type="Enumeration" value="Query"/>
0043	<RequestPayload>
0044	<QueryFunction type="Enumeration" value="QueryOperations"/>
0045	<QueryFunction type="Enumeration" value="QueryObjects"/>
0046	</RequestPayload>
0047	</BatchItem>
0048	</RequestMessage>
	42007801000000904200770100000048420069010000002042006a02000000040000000100000000 42006b020000000400000000000000004200500200000004000008000000000042000d0200000004 000000010000000042000f010000003842005c050000000400000018000000004200790100000020 4200740500000004000000010000000042007405000000040000000200000000
0049	<ResponseMessage>
0050	<ResponseHeader>
0051	<ProtocolVersion>
0052	<ProtocolVersionMajor type="Integer" value="1"/>
0053	<ProtocolVersionMinor type="Integer" value="0"/>
0054	</ProtocolVersion>
0055	<TimeStamp type="DateTime" value="2013-06-26T09:09:17+00:00"/>
0056	<BatchCount type="Integer" value="1"/>
0057	</ResponseHeader>
0058	<BatchItem>
0059	<Operation type="Enumeration" value="Query"/>
0060	<ResultStatus type="Enumeration" value="Success"/>
0061	<ResponsePayload>
0062	<Operation type="Enumeration" value="Query"/>
0063	<Operation type="Enumeration" value="Locate"/>
0064	<Operation type="Enumeration" value="Destroy"/>
0065	<Operation type="Enumeration" value="Get"/>
0066	<Operation type="Enumeration" value="Create"/>

0067	<Operation type="Enumeration" value="Register"/>
0068	<Operation type="Enumeration" value="GetAttributes"/>
0069	<Operation type="Enumeration" value="GetAttributeList"/>
0070	<Operation type="Enumeration" value="AddAttribute"/>
0071	<Operation type="Enumeration" value="ModifyAttribute"/>
0072	<Operation type="Enumeration" value="DeleteAttribute"/>
0073	<Operation type="Enumeration" value="Activate"/>
0074	<Operation type="Enumeration" value="Revoke"/>
0075	<Operation type="Enumeration" value="Poll"/>
0076	<Operation type="Enumeration" value="Cancel"/>
0077	<Operation type="Enumeration" value="Check"/>
0078	<Operation type="Enumeration" value="GetUsageAllocation"/>
0079	<Operation type="Enumeration" value="CreateKeyPair"/>
0080	<Operation type="Enumeration" value="ReKey"/>
0081	<Operation type="Enumeration" value="Archive"/>
0082	<Operation type="Enumeration" value="Recover"/>
0083	<Operation type="Enumeration" value="ObtainLease"/>
0084	<Operation type="Enumeration" value="Certify"/>
0085	<Operation type="Enumeration" value="ReCertify"/>
0086	<Operation type="Enumeration" value="Notify"/>
0087	<Operation type="Enumeration" value="Put"/>
0088	<ObjectType type="Enumeration" value="Certificate"/>
0089	<ObjectType type="Enumeration" value="SymmetricKey"/>
0090	<ObjectType type="Enumeration" value="SecretData"/>
0091	<ObjectType type="Enumeration" value="PublicKey"/>
0092	<ObjectType type="Enumeration" value="PrivateKey"/>
0093	<ObjectType type="Enumeration" value="Template"/>
0094	<ObjectType type="Enumeration" value="OpaqueObject"/>
0095	<ObjectType type="Enumeration" value="SplitKey"/>
0096	</ResponsePayload>
0097	</BatchItem>
0098	</ResponseMessage>
	42007b01000002a042007a0100000048420069010000002042006a02000000040000000100000000
	42006b0200000004000000000000000042009209000000080000000051caaafbd42000d0200000004
	000000010000000042000f010000024842005c0500000004000000180000000042007f0500000004
	000000000000000042007c010000022042005c0500000004000000180000000042005c0500000004
	000000080000000042005c0500000004000000140000000042005c05000000040000000a00000000
	42005c0500000004000000010000000042005c0500000004000000030000000042005c0500000004
	0000000b0000000042005c05000000040000000c0000000042005c05000000040000000d00000000
	42005c05000000040000000e0000000042005c05000000040000000f0000000042005c0500000004
	000000120000000042005c0500000004000000130000000042005c05000000040000001a00000000
	42005c0500000004000000190000000042005c0500000004000000090000000042005c0500000004
	000000110000000042005c0500000004000000020000000042005c05000000040000000400000000
	42005c0500000004000000150000000042005c0500000004000000160000000042005c0500000004
	000000100000000042005c0500000004000000060000000042005c05000000040000000700000000
	42005c05000000040000001b0000000042005c05000000040000001c000000004200570500000004
	00000001000000004200570500000004000000020000000042005705000000040000000700000000
	42005705000000040000000300000000420057050000000400000004000000004200570500000004
	00000006000000004200570500000004000000080000000042005705000000040000000500000000

538

## 539 7.2 Mandatory XML Profile Test Cases KMIP v1.1

### 540 7.2.1 MSGENC-XML-M-1-11 - Query, Maximum Response Size

541 Perform a Query operation, querying the Operations and Objects supported by the server, with a  
542 restriction on the Maximum Response Size set in the request header. Since the resulting Query response

- 543 is too big, an error is returned. Increase the Maximum Response Size, resubmit the Query request, and  
544 get a successful response.  
545 The specific list of operations and object types returned in the response MAY vary.

0001	# TIME 0
0002	<RequestMessage>
0003	<RequestHeader>
0004	<ProtocolVersion>
0005	<ProtocolVersionMajor type="Integer" value="1"/>
0006	<ProtocolVersionMinor type="Integer" value="1"/>
0007	</ProtocolVersion>
0008	<MaximumResponseSize type="Integer" value="256"/>
0009	<BatchCount type="Integer" value="1"/>
0010	</RequestHeader>
0011	<BatchItem>
0012	<Operation type="Enumeration" value="Query"/>
0013	<RequestPayload>
0014	<QueryFunction type="Enumeration" value="QueryOperations"/>
0015	<QueryFunction type="Enumeration" value="QueryObjects"/>
0016	</RequestPayload>
0017	</BatchItem>
0018	</RequestMessage>
0019	42007801000000904200770100000048420069010000002042006a02000000040000000100000000
0020	42006b0200000004000000001000000004200500200000004000001000000000042000d0200000004
0021	000000010000000042000f010000003842005c050000000400000018000000004200790100000020
0022	4200740500000004000000010000000042007405000000040000000200000000
0023	<ResponseMessage>
0024	<ResponseHeader>
0025	<ProtocolVersion>
0026	<ProtocolVersionMajor type="Integer" value="1"/>
0027	<ProtocolVersionMinor type="Integer" value="1"/>
0028	</ProtocolVersion>
0029	<TimeStamp type="DateTime" value="2014-06-10T08:03:34+00:00"/>
0030	<BatchCount type="Integer" value="1"/>
0031	</ResponseHeader> <BatchItem>
0032	<Operation type="Enumeration" value="Query"/>
0033	<ResultStatus type="Enumeration" value="OperationFailed"/>
0034	<ResultReason type="Enumeration" value="ResponseTooLarge"/>
0035	<ResultMessage type="TextString" value="TOO_LARGE"/>
0036	</BatchItem>
0037	</ResponseMessage>
0038	42007b01000000a042007a0100000048420069010000002042006a02000000040000000100000000
0039	42006b02000000040000000010000000042009209000000080000000005396bc2442000d0200000004
0040	000000010000000042000f010000004842005c0500000004000000180000000042007f0500000004
0041	000000010000000042007e0500000004000000020000000042007d0700000009544f4f5f4c415247
0042	4500000000000000
0043	# TIME 1
0044	<RequestMessage>
0045	<RequestHeader>
0046	<ProtocolVersion>
0047	<ProtocolVersionMajor type="Integer" value="1"/>
0048	<ProtocolVersionMinor type="Integer" value="1"/>
0049	</ProtocolVersion>
0050	<MaximumResponseSize type="Integer" value="2048"/>
0051	<BatchCount type="Integer" value="1"/>
0052	</RequestHeader>



0042	<BatchItem>
0043	<Operation type="Enumeration" value="Query"/>
0044	<RequestPayload>
0045	<QueryFunction type="Enumeration" value="QueryOperations"/>
0046	<QueryFunction type="Enumeration" value="QueryObjects"/>
0047	</RequestPayload>
0048	</BatchItem>
0049	</RequestMessage>
	42007801000000904200770100000048420069010000002042006a02000000040000000100000000 42006b0200000004000000001000000004200500200000004000008000000000042000d0200000004 000000010000000042000f010000003842005c050000000400000018000000004200790100000020 42007405000000040000000010000000042007405000000040000000200000000
0050	<ResponseMessage>
0051	<ResponseHeader>
0052	<ProtocolVersion>
0053	<ProtocolVersionMajor type="Integer" value="1"/>
0054	<ProtocolVersionMinor type="Integer" value="1"/>
0055	</ProtocolVersion>
0056	<TimeStamp type="DateTime" value="2014-06-10T08:03:34+00:00"/>
0057	<BatchCount type="Integer" value="1"/>
0058	</ResponseHeader>
0059	<BatchItem>
0060	<Operation type="Enumeration" value="Query"/>
0061	<ResultStatus type="Enumeration" value="Success"/>
0062	<ResponsePayload>
0063	<Operation type="Enumeration" value="Query"/>
0064	<Operation type="Enumeration" value="Locate"/>
0065	<Operation type="Enumeration" value="Destroy"/>
0066	<Operation type="Enumeration" value="Get"/>
0067	<Operation type="Enumeration" value="Create"/>
0068	<Operation type="Enumeration" value="Register"/>
0069	<Operation type="Enumeration" value="GetAttributes"/>
0070	<Operation type="Enumeration" value="GetAttributeList"/>
0071	<Operation type="Enumeration" value="AddAttribute"/>
0072	<Operation type="Enumeration" value="ModifyAttribute"/>
0073	<Operation type="Enumeration" value="DeleteAttribute"/>
0074	<Operation type="Enumeration" value="Activate"/>
0075	<Operation type="Enumeration" value="Revoke"/>
0076	<Operation type="Enumeration" value="Poll"/>
0077	<Operation type="Enumeration" value="Cancel"/>
0078	<Operation type="Enumeration" value="Check"/>
0079	<Operation type="Enumeration" value="GetUsageAllocation"/>
0080	<Operation type="Enumeration" value="CreateKeyPair"/>
0081	<Operation type="Enumeration" value="ReKey"/>
0082	<Operation type="Enumeration" value="Archive"/>
0083	<Operation type="Enumeration" value="Recover"/>
0084	<Operation type="Enumeration" value="ObtainLease"/>
0085	<Operation type="Enumeration" value="ReKeyKeyPair"/>
0086	<Operation type="Enumeration" value="Certify"/>
0087	<Operation type="Enumeration" value="ReCertify"/>
0088	<Operation type="Enumeration" value="DiscoverVersions"/>
0089	<Operation type="Enumeration" value="Notify"/>
0090	<Operation type="Enumeration" value="Put"/>
0091	<ObjectType type="Enumeration" value="Certificate"/>
0092	<ObjectType type="Enumeration" value="SymmetricKey"/>
0093	<ObjectType type="Enumeration" value="SecretData"/>
0094	<ObjectType type="Enumeration" value="PublicKey"/>



0095	<ObjectType type="Enumeration" value="PrivateKey"/>
0096	<ObjectType type="Enumeration" value="Template"/>
0097	<ObjectType type="Enumeration" value="OpaqueObject"/>
0098	<ObjectType type="Enumeration" value="SplitKey"/>
0099	</ResponsePayload>
0100	</BatchItem>
0101	</ResponseMessage>
	42007b010000002c042007a0100000048420069010000002042006a02000000040000000100000000 42006b0200000004000000001000000004200920900000008000000005396bc2442000d0200000004 000000010000000042000f010000026842005c0500000004000000180000000042007f0500000004 000000000000000042007c010000024042005c0500000004000000180000000042005c0500000004 000000080000000042005c0500000004000000140000000042005c05000000040000000a00000000 42005c0500000004000000010000000042005c05000000040000000030000000042005c0500000004 0000000b0000000042005c05000000040000000c0000000042005c05000000040000000d00000000 42005c05000000040000000e0000000042005c05000000040000000f0000000042005c0500000004 000000120000000042005c0500000004000000130000000042005c05000000040000001a00000000 42005c0500000004000000190000000042005c0500000004000000090000000042005c0500000004 000000110000000042005c0500000004000000200000000042005c05000000040000000400000000 42005c0500000004000000150000000042005c0500000004000000160000000042005c0500000004 000000100000000042005c05000000040000001d0000000042005c05000000040000000600000000 42005c0500000004000000070000000042005c05000000040000001e0000000042005c0500000004 0000001b0000000042005c05000000040000001c0000000042005705000000040000000100000000 42005705000000040000000200000000420057050000000400000007000000004200570500000004 00000003000000004200570500000004000000040000000042005705000000040000000600000000 4200570500000004000000080000000042005705000000040000000500000000

546

## 547 7.3 Mandatory XML Profile Test Cases KMIP v1.2

### 548 7.3.1 MSGENC-XML-M-1-12 - Query, Maximum Response Size

549 Perform a Query operation, querying the Operations and Objects supported by the server, with a  
550 restriction on the Maximum Response Size set in the request header. Since the resulting Query response  
551 is too big, an error is returned. Increase the Maximum Response Size, resubmit the Query request, and  
552 get a successful response.

553 The specific list of operations and object types returned in the response MAY vary.

	# TIME 0
0001	<RequestMessage>
0002	<RequestHeader>
0003	<ProtocolVersion>
0004	<ProtocolVersionMajor type="Integer" value="1"/>
0005	<ProtocolVersionMinor type="Integer" value="2"/>
0006	</ProtocolVersion>
0007	<MaximumResponseSize type="Integer" value="256"/>
0008	<BatchCount type="Integer" value="1"/>
0009	</RequestHeader>
0010	<BatchItem>
0011	<Operation type="Enumeration" value="Query"/>
0012	<RequestPayload>
0013	<QueryFunction type="Enumeration" value="QueryOperations"/>
0014	<QueryFunction type="Enumeration" value="QueryObjects"/>
0015	</RequestPayload>
0016	</BatchItem>
0017	</RequestMessage>
	42007801000000904200770100000048420069010000002042006a02000000040000000100000000 42006b020000000400000002000000004200500200000004000001000000000042000d0200000004 000000010000000042000f01000003842005c050000000400000018000000004200790100000020 4200740500000004000000010000000042007405000000040000000200000000

0018	<ResponseMessage>
0019	<ResponseHeader>
0020	<ProtocolVersion>
0021	<ProtocolVersionMajor type="Integer" value="1"/>
0022	<ProtocolVersionMinor type="Integer" value="2"/>
0023	</ProtocolVersion>
0024	<TimeStamp type="DateTime" value="2014-06-10T08:07:28+00:00"/>
0025	<BatchCount type="Integer" value="1"/>
0026	</ResponseHeader> <BatchItem>
0027	<Operation type="Enumeration" value="Query"/>
0028	<ResultStatus type="Enumeration" value="OperationFailed"/>
0029	<ResultReason type="Enumeration" value="ResponseTooLarge"/>
0030	<ResultMessage type="TextString" value="TOO_LARGE"/>
0031	</BatchItem>
0032	</ResponseMessage>
	42007b01000000a042007a0100000048420069010000002042006a02000000040000000100000000 42006b0200000004000000002000000004200920900000008000000005396bcc042000d0200000004 000000010000000042000f010000004842005c0500000004000000180000000042007f0500000004 000000010000000042007e0500000004000000020000000042007d0700000009544f4f5f4c415247 4500000000000000
	# TIME 1
0033	<RequestMessage>
0034	<RequestHeader>
0035	<ProtocolVersion>
0036	<ProtocolVersionMajor type="Integer" value="1"/>
0037	<ProtocolVersionMinor type="Integer" value="2"/>
0038	</ProtocolVersion>
0039	<MaximumResponseSize type="Integer" value="2048"/>
0040	<BatchCount type="Integer" value="1"/>
0041	</RequestHeader>
0042	<BatchItem>
0043	<Operation type="Enumeration" value="Query"/>
0044	<RequestPayload>
0045	<QueryFunction type="Enumeration" value="QueryOperations"/>
0046	<QueryFunction type="Enumeration" value="QueryObjects"/>
0047	</RequestPayload>
0048	</BatchItem>
0049	</RequestMessage>
	42007801000000904200770100000048420069010000002042006a02000000040000000100000000 42006b020000000400000000200000000420050020000000400000800000000042000d0200000004 000000010000000042000f010000003842005c050000000400000018000000004200790100000020 4200740500000004000000010000000042007405000000040000000200000000
0050	<ResponseMessage>
0051	<ResponseHeader>
0052	<ProtocolVersion>
0053	<ProtocolVersionMajor type="Integer" value="1"/>
0054	<ProtocolVersionMinor type="Integer" value="2"/>
0055	</ProtocolVersion>
0056	<TimeStamp type="DateTime" value="2014-06-10T08:07:28+00:00"/>
0057	<BatchCount type="Integer" value="1"/>
0058	</ResponseHeader>
0059	<BatchItem>
0060	<Operation type="Enumeration" value="Query"/>
0061	<ResultStatus type="Enumeration" value="Success"/>

0062	<ResponsePayload>
0063	<Operation type="Enumeration" value="Query"/>
0064	<Operation type="Enumeration" value="Locate"/>
0065	<Operation type="Enumeration" value="Destroy"/>
0066	<Operation type="Enumeration" value="Get"/>
0067	<Operation type="Enumeration" value="Create"/>
0068	<Operation type="Enumeration" value="Register"/>
0069	<Operation type="Enumeration" value="GetAttributes"/>
0070	<Operation type="Enumeration" value="GetAttributeList"/>
0071	<Operation type="Enumeration" value="AddAttribute"/>
0072	<Operation type="Enumeration" value="ModifyAttribute"/>
0073	<Operation type="Enumeration" value="DeleteAttribute"/>
0074	<Operation type="Enumeration" value="Activate"/>
0075	<Operation type="Enumeration" value="Revoke"/>
0076	<Operation type="Enumeration" value="Poll"/>
0077	<Operation type="Enumeration" value="Cancel"/>
0078	<Operation type="Enumeration" value="Check"/>
0079	<Operation type="Enumeration" value="GetUsageAllocation"/>
0080	<Operation type="Enumeration" value="CreateKeyPair"/>
0081	<Operation type="Enumeration" value="ReKey"/>
0082	<Operation type="Enumeration" value="Archive"/>
0083	<Operation type="Enumeration" value="Recover"/>
0084	<Operation type="Enumeration" value="ObtainLease"/>
0085	<Operation type="Enumeration" value="ReKeyKeyPair"/>
0086	<Operation type="Enumeration" value="Certify"/>
0087	<Operation type="Enumeration" value="ReCertify"/>
0088	<Operation type="Enumeration" value="DiscoverVersions"/>
0089	<Operation type="Enumeration" value="Notify"/>
0090	<Operation type="Enumeration" value="Put"/>
0091	<Operation type="Enumeration" value="RNGRetrieve"/>
0092	<Operation type="Enumeration" value="RNGSeed"/>
0093	<Operation type="Enumeration" value="Encrypt"/>
0094	<Operation type="Enumeration" value="Decrypt"/>
0095	<Operation type="Enumeration" value="Sign"/>
0096	<Operation type="Enumeration" value="SignatureVerify"/>
0097	<Operation type="Enumeration" value="MAC"/>
0098	<Operation type="Enumeration" value="MACVerify"/>
0099	<Operation type="Enumeration" value="Hash"/>
0100	<Operation type="Enumeration" value="CreateSplitKey"/>
0101	<Operation type="Enumeration" value="JoinSplitKey"/>
0102	<ObjectType type="Enumeration" value="Certificate"/>
0103	<ObjectType type="Enumeration" value="SymmetricKey"/>
0104	<ObjectType type="Enumeration" value="SecretData"/>
0105	<ObjectType type="Enumeration" value="PublicKey"/>
0106	<ObjectType type="Enumeration" value="PrivateKey"/>
0107	<ObjectType type="Enumeration" value="Template"/>
0108	<ObjectType type="Enumeration" value="OpaqueObject"/>
0109	<ObjectType type="Enumeration" value="SplitKey"/>
0110	<ObjectType type="Enumeration" value="PGPKey"/>
0111	</ResponsePayload>
0112	</BatchItem>
0113	</ResponseMessage>
	42007b010000038042007a0100000048420069010000002042006a02000000040000000100000000
	42006b0200000004000000002000000004200920900000008000000005396bcc042000d0200000004
	000000010000000042000f010000032842005c0500000004000000180000000042007f0500000004
	000000000000000042007c010000030042005c0500000004000000180000000042005c0500000004
	000000080000000042005c0500000004000000140000000042005c05000000040000000a00000000
	42005c0500000004000000010000000042005c0500000004000000030000000042005c0500000004

0000000b0000000042005c05000000040000000c0000000042005c05000000040000000d00000000
42005c05000000040000000e0000000042005c05000000040000000f0000000042005c0500000004
000000120000000042005c0500000004000000130000000042005c05000000040000001a00000000
42005c0500000004000000190000000042005c0500000004000000090000000042005c0500000004
000000110000000042005c0500000004000000020000000042005c05000000040000000400000000
42005c0500000004000000150000000042005c0500000004000000160000000042005c0500000004
000000100000000042005c05000000040000001d0000000042005c05000000040000000600000000
42005c0500000004000000070000000042005c05000000040000001e0000000042005c0500000004
0000001b0000000042005c05000000040000001c0000000042005c05000000040000002500000000
42005c0500000004000000260000000042005c05000000040000001f0000000042005c0500000004
000000200000000042005c0500000004000000210000000042005c05000000040000002200000000
42005c0500000004000000230000000042005c0500000004000000240000000042005c0500000004
000000270000000042005c0500000004000000280000000042005c05000000040000002900000000
42005705000000040000000100000000420057050000000400000002000000004200570500000004
000000700000000042005705000000040000000300000000042005705000000040000000400000000
42005705000000040000000600000000420057050000000400000008000000004200570500000004
000000500000000042005705000000040000000900000000

---

## 8 Conformance

### 8.1 HTTPS Profile

#### 8.1.1 HTTPS Client KMIP v1.0 Profile Conformance

KMIP client implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions as specified in Section 2.1 of this profile.
2. SHALL support the KMIP Port Number conditions as specified in Section 2.2 of this profile.
3. SHALL support the Request URL conditions as specified in Section 2.3 of this profile.
4. SHALL support the HTTP Encoding conditions as specified in Section 2.4 of this profile.
5. SHALL support all the Mandatory HTTPS Profile Test Cases KMIP v1.0 (3.1)

#### 8.1.2 HTTPS Client KMIP v1.1 Profile Conformance

KMIP client implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions as specified in Section 2.1 of this profile.
2. SHALL support the KMIP Port Number conditions as specified in Section 2.2 of this profile.
3. SHALL support the Request URL conditions as specified in Section 2.3 of this profile.
4. SHALL support the HTTP Encoding conditions as specified in Section 2.4 of this profile.
5. SHALL support all the Mandatory HTTPS Profile Test Cases KMIP v1.1 (3.2)

#### 8.1.3 HTTPS Client KMIP v1.2 Profile Conformance

KMIP client implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions as specified in Section 2.1 of this profile.
2. SHALL support the KMIP Port Number conditions as specified in Section 2.2 of this profile.
3. SHALL support the Request URL conditions as specified in Section 2.3 of this profile.
4. SHALL support the HTTP Encoding conditions as specified in Section 2.4 of this profile.
5. SHALL support all the Mandatory HTTPS Profile Test Cases KMIP v1.2 (3.3)

#### 8.1.4 HTTPS Server KMIP v1.0 Profile Conformance

KMIP server implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions as specified in Section 2.1 of this profile.
2. SHALL support the KMIP Port Number conditions as specified in Section 2.2 of this profile.
3. SHALL support the Request URL conditions as specified in Section 2.3 of this profile.
4. SHALL support the HTTP Encoding conditions as specified in Section 2.5 of this profile.
5. SHALL support all the Mandatory HTTPS Profile Test Cases KMIP v1.0 (3.1)

#### 8.1.5 HTTPS Server KMIP v1.1 Profile Conformance

KMIP server implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions as specified in Section 2.1 of this profile.
2. SHALL support the KMIP Port Number conditions as specified in Section 2.2 of this profile.
3. SHALL support the Request URL conditions as specified in Section 2.3 of this profile.

- 590 4. SHALL support the HTTP Encoding conditions as specified in Section 2.5 of this profile.  
591 5. Mandatory HTTPS Profile Test Cases KMIP v1.1 (3.2)

## 592 **8.1.6 HTTPS Server KMIP v1.2 Profile Conformance**

593 KMIP server implementations conformant to this profile:

- 594 1. SHALL support the Authentication Suite conditions as specified in Section 2.1 of this profile.  
595 2. SHALL support the KMIP Port Number conditions as specified in Section 2.2 of this profile.  
596 3. SHALL support the Request URL conditions as specified in Section 2.3 of this profile.  
597 4. SHALL support the HTTP Encoding conditions as specified in Section 2.5 of this profile.  
598 5. SHALL support all the Mandatory HTTPS Profile Test Cases KMIP v1.2 (3.3)

## 599 **8.2 JSON Profile**

### 600 **8.2.1 JSON Client KMIP v1.0 Profile Conformance**

601 KMIP client implementations conformant to this profile:

- 602 1. SHALL support JSON message encoding for request and response messages as specified in  
603 Section 4.1 of this profile.  
604 2. SHALL support all the Mandatory JSON Profile Test Cases KMIP v1.0 (5.1)  
605 3. SHALL support mapping of all TTLV tags and enumerations specified within each version of the  
606 [KMIP-SPEC] that is supported  
607 4. SHALL support user defined extensions containing additional tags and enumerations not  
608 specified within [KMIP-SPEC]

### 609 **8.2.2 JSON Client KMIP v1.1 Profile Conformance**

610 KMIP client implementations conformant to this profile:

- 611 1. SHALL support JSON message encoding for request and response messages as specified in  
612 Section 4.1 of this profile.  
613 2. SHALL support all the Mandatory JSON Profile Test Cases KMIP v1.1 (5.2)  
614 3. SHALL support mapping of all TTLV tags and enumerations specified within each version of the  
615 [KMIP-SPEC] that is supported  
616 4. SHALL support user defined extensions containing additional tags and enumerations not  
617 specified within [KMIP-SPEC]

### 618 **8.2.3 JSON Client KMIP v1.2 Profile Conformance**

619 KMIP client implementations conformant to this profile:

- 620 1. SHALL support JSON message encoding for request and response messages as specified in  
621 Section 4.1 of this profile.  
622 2. SHALL support all the Mandatory JSON Profile Test Cases KMIP v1.2(5.3)  
623 3. SHALL support mapping of all TTLV tags and enumerations specified within each version of the  
624 [KMIP-SPEC] that is supported  
625 4. SHALL support user defined extensions containing additional tags and enumerations not  
626 specified within [KMIP-SPEC]

### 627 **8.2.4 JSON Server KMIP v1.0 Profile Conformance**

628 KMIP server implementations conformant to this profile:

- 629 1. SHALL support JSON message encoding for request and response messages as specified in  
630 Section 4.1 of this profile.
- 631 2. SHALL support all the Mandatory JSON Profile Test Cases KMIP v1.0 (5.1)
- 632 3. SHALL support mapping of all TTLV tags and enumerations specified within each version of the  
633 [KMIP-SPEC] that is supported
- 634 4. SHALL support user defined extensions containing additional tags and enumerations not  
635 specified within [KMIP-SPEC]

## 636 8.2.5 JSON Server KMIP v1.1 Profile Conformance

637 KMIP server implementations conformant to this profile:

- 638 1. SHALL support JSON message encoding for request and response messages as specified in  
639 Section 4.1 of this profile.
- 640 2. SHALL support all the Mandatory JSON Profile Test Cases KMIP v1.1 (5.2)
- 641 3. SHALL support mapping of all TTLV tags and enumerations specified within each version of the  
642 [KMIP-SPEC] that is supported
- 643 4. SHALL support user defined extensions containing additional tags and enumerations not  
644 specified within [KMIP-SPEC]

## 645 8.2.6 JSON Server KMIP v1.2 Profile Conformance

646 KMIP server implementations conformant to this profile:

- 647 1. SHALL support JSON message encoding for request and response messages as specified in  
648 Section 4.1 of this profile.
- 649 2. SHALL support all the Mandatory JSON Profile Test Cases KMIP v1.2(5.3)
- 650 3. SHALL support mapping of all TTLV tags and enumerations specified within each version of the  
651 [KMIP-SPEC] that is supported
- 652 4. SHALL support user defined extensions containing additional tags and enumerations not  
653 specified within [KMIP-SPEC]

## 654 8.3 XML Profile

### 655 8.3.1 XML Client KMIP v1.0 Profile Conformance

656 KMIP client implementations conformant to this profile:

- 657 1. SHALL support XML message encoding for request and response messages as specified in  
658 Section 6.1 of this profile.
- 659 2. SHALL support all the Mandatory XML Profile Test Cases KMIP v1.0(7.1)
- 660 3. SHALL support mapping of all TTLV tags and enumerations specified within each version of the  
661 [KMIP-SPEC] that is supported.
- 662 4. SHALL support user defined extensions containing additional tags and enumerations not  
663 specified within [KMIP-SPEC].

### 664 8.3.2 XML Client KMIP v1.1 Profile Conformance

665 KMIP client implementations conformant to this profile:

- 666 1. SHALL support XML message encoding for request and response messages as specified in  
667 Section 6.1 of this profile.
- 668 2. SHALL support all the Mandatory XML Profile Test Cases KMIP v1.1(7.2)
- 669 3. SHALL support mapping of all TTLV tags and enumerations specified within each version of the  
670 [KMIP-SPEC] that is supported.



- 671 4. SHALL support user defined extensions containing additional tags and enumerations not  
672 specified within [KMIP-SPEC].

### 673 **8.3.3 XML Client KMIP v1.2 Profile Conformance**

674 KMIP client implementations conformant to this profile:

- 675 1. SHALL support XML message encoding for request and response messages as specified in  
676 Section 6.1 of this profile.
- 677 2. SHALL support all the Mandatory XML Profile Test Cases KMIP v1.2(7.3)
- 678 3. SHALL support mapping of all TTLV tags and enumerations specified within each version of the  
679 [KMIP-SPEC] that is supported.
- 680 4. SHALL support user defined extensions containing additional tags and enumerations not  
681 specified within [KMIP-SPEC].

### 682 **8.3.4 XML Server KMIP v1.0 Profile Conformance**

683 KMIP server implementations conformant to this profile:

- 684 1. SHALL support XML message encoding for request and response messages as specified in  
685 Section 6.1 of this profile.
- 686 2. SHALL support all the Mandatory XML Profile Test Cases KMIP v1.0(7.1)
- 687 3. SHALL support mapping of all TTLV tags and enumerations specified within each version of the  
688 [KMIP-SPEC] that is supported.
- 689 4. SHALL support user defined extensions containing additional tags and enumerations not  
690 specified within [KMIP-SPEC].

### 691 **8.3.5 XML Server KMIP v1.1 Profile Conformance**

692 KMIP server implementations conformant to this profile:

- 693 1. SHALL support XML message encoding for request and response messages as specified in  
694 Section 6.1 of this profile.
- 695 2. SHALL support all the Mandatory XML Profile Test Cases KMIP v1.1(7.2)
- 696 3. SHALL support mapping of all TTLV tags and enumerations specified within each version of the  
697 [KMIP-SPEC] that is supported.
- 698 4. SHALL support user defined extensions containing additional tags and enumerations not  
699 specified within [KMIP-SPEC].

### 700 **8.3.6 XML Server KMIP v1.2 Profile Conformance**

701 KMIP server implementations conformant to this profile:

- 702 1. SHALL support XML message encoding for request and response messages as specified in  
703 Section 6.1 of this profile.
- 704 2. SHALL support all the Mandatory XML Profile Test Cases KMIP v1.2(7.3)
- 705 3. SHALL support mapping of all TTLV tags and enumerations specified within each version of the  
706 [KMIP-SPEC] that is supported.
- 707 4. SHALL support user defined extensions containing additional tags and enumerations not  
708 specified within [KMIP-SPEC].

## 709 **8.4 Permitted Test Case Variations**

710 Whilst the test cases provided in this Profile define the allowed request and response content, some  
711 inherent variations MAY occur and are permitted within a successfully completed test case.



Each test case MAY include allowed variations in the description of the test case in addition to the variations noted in this section.

Other variations not explicitly noted in this Profile SHALL be deemed non-conformant.

#### 8.4.1 Variable Items

An implementation conformant to this Profile MAY vary the following values:

1. UniqueIdentifier
2. PrivateKeyUniqueIdentifier
3. PublicKeyUniqueIdentifier
4. UniqueBatchItemIdentifier
5. AsynchronousCorrelationValue
6. TimeStamp
7. KeyValue / KeyMaterial including:
  - a. key material content returned for managed cryptographic objects which are generated by the server
  - b. wrapped versions of keys where the wrapping key is dynamic or the wrapping contains variable output for each wrap operation
8. For response containing the output of cryptographic operation in Data / SignatureData/ MACData / IVCounterNonce where:
  - a. the managed object is generated by the server; or
  - b. the operation inherently contains variable output
9. For the following DateTime attributes where the value is not specified in the request as a fixed DateTime value:
  - a. ActivationDate
  - b. ArchiveDate
  - c. CompromiseDate
  - d. CompromiseOccurrenceDate
  - e. DeactivationDate
  - f. DestroyDate
  - g. InitialDate
  - h. LastChangeDate
  - i. ProtectStartDate
  - j. ProcessStopDate
  - k. ValidityDate
  - l. OriginalCreationDate
10. LinkedObjectIdentifier
11. DigestValue
  - a. For those managed cryptographic objects which are dynamically generated
12. KeyFormatType
  - a. The key format type selected by the server when it creates managed objects
13. Digest
  - a. The HashingAlgorithm selected by the server when it calculates the digest for a managed object for which it has access to the key material
  - b. The Digest Value

- 14. Extensions reported in Query for ExtensionList and ExtensionMap
- 15. Application Namespaces reported in Query
- 16. Object Types reported in Query other than those noted as required in this profile
- 17. Operation Types reported in Query other than those noted as required in this profile (or any referenced profile documents)
- 18. For TextString attribute values containing test identifiers:
  - a. Additional vendor or application prefixes
- 19. Additional attributes beyond those noted in the response

An implementation conformant to this Profile MAY allow the following response variations:

- 20. Object Group values – May or may not return one or more Object Group values not included in the requests
- 21. y-CustomAttributes – May or may not include additional server-specific associated attributes not included in requests
- 22. Message Extensions – May or may not include additional (non-critical) vendor extensions
- 23. TemplateAttribute – May or may not be included in responses where the Template Attribute response is noted as optional in [KMIP-SPEC]
- 24. AttributeIndex – May or may not include Attribute Index value where the Attribute Index value is 0 for Protocol Versions 1.1 and above.
- 25. ResultMessage – May or may not be included in responses and the value (if included) may vary from the text contained within the test case.
- 26. The list of Protocol Versions returned in a DiscoverVersion response may include additional protocol versions if the request has not specified a list of client supported Protocol Versions.
- 27. VendorIdentification - The value (if included) may vary from the text contained within the test case.

## 8.4.2 Variable behavior

An implementation conformant to this Profile SHALL allow variation of the following behavior:

- 1. A test MAY omit the clean-up requests and responses (containing Revoke and/or Destroy) at the end of the test provided there is a separate mechanism to remove the created objects during testing.
- 2. A test MAY omit the test identifiers if the client is unable to include them in requests. This includes the following attributes:
  - a. Name; and
  - b. x-ID
- 3. A test MAY perform requests with multiple batch items or as multiple requests with a single batch item provided the sequence of operations are equivalent
- 4. A request MAY contain an optional *Authentication* [KMIP\_SPEC] structure within each request.

---

## Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

### Original HTTPS Profile Proposal:

Alan Frindell, SafeNet, Inc.

### Original HTTPS Profile Contributors:

Mathias Björkqvist, IBM

### Participants:

Hal Aldridge, Sypris Electronics  
Mike Allen, Symantec  
Gordon Arnold, IBM  
Todd Arnold, IBM  
Richard Austin, Hewlett-Packard  
Lars Bagnert, PrimeKey  
Elaine Barker, NIST  
Peter Bartok, Venafi, Inc.  
Tom Benjamin, IBM  
Anthony Berglas, Cryptsoft  
Mathias Björkqvist, IBM  
Kevin Bocket, Venafi  
Anne Bolgert, IBM  
Alan Brown, Thales e-Security  
Tim Bruce, CA Technologies  
Chris Burchett, Credant Technologies, Inc.  
Kelley Burgin, National Security Agency  
Robert Burns, Thales e-Security  
Chuck Castleton, Venafi  
Kenli Chong, QuintessenceLabs  
John Clark, Hewlett-Packard  
Tom Clifford, Symantec Corp.  
Doron Cohen, SafeNet, Inc.  
Tony Cox, Cryptsoft  
Russell Dietz, SafeNet, Inc.  
Graydon Dodson, Lexmark International Inc.  
Vinod Duggirala, EMC Corporation  
Chris Dunn, SafeNet, Inc.  
Michael Duren, Sypris Electronics  
James Dzierzanowski, American Express CCoE  
Faisal Faruqui, Thales e-Security  
Stan Feather, Hewlett-Packard  
David Finkelstein, Symantec Corp.  
James Fitzgerald, SafeNet, Inc.  
Indra Fitzgerald, Hewlett-Packard  
Judith Furlong, EMC Corporation  
Susan Gleeson, Oracle  
Robert Griffin, EMC Corporation  
Paul Grojean, Individual  
Robert Haas, IBM  
Thomas Hardjono, M.I.T.  
ChengDong He, Huawei Technologies Co., Ltd.  
Steve He, Vormetric

841 Kurt Heberlein, Hewlett-Packard  
842 Larry Hofer, Emulex Corporation  
843 Maryann Hondo, IBM  
844 Walt Hubis, NetApp  
845 Tim Hudson, Cryptsoft  
846 Jonas Iggbom, Venafi, Inc.  
847 Sitaram Inguva, American Express CCoE  
848 Jay Jacobs, Target Corporation  
849 Glen Jaquette, IBM  
850 Mahadev Karadiguddi, NetApp  
851 Greg Kazmierczak, Wave Systems Corp.  
852 Marc Kenig, SafeNet, Inc.  
853 Mark Knight, Thales e-Security  
854 Kathy Kriese, Symantec Corporation  
855 Mark Lambiase, SecureAuth  
856 John Leiseboer, Quintessence Labs  
857 Hal Lockhart, Oracle Corporation  
858 Robert Lockhart, Thales e-Security  
859 Anne Luk, Cryptsoft  
860 Sairam Manidi, Freescale  
861 Luther Martin, Voltage Security  
862 Neil McEvoy, iFOSSF  
863 Marina Milshtein, Individual  
864 Dale Moberg, Axway Software  
865 Jishnu Mukeri, Hewlett-Packard  
866 Bryan Olson, Hewlett-Packard  
867 John Peck, IBM  
868 Rob Philpott, EMC Corporation  
869 Denis Pochuev, SafeNet, Inc.  
870 Reid Poole, Venafi, Inc.  
871 Ajai Puri, SafeNet, Inc.  
872 Saravanan Ramalingam, Thales e-Security  
873 Peter Reed, SafeNet, Inc.  
874 Bruce Rich, IBM  
875 Christina Richards, American Express CCoE  
876 Warren Robbins, Dell  
877 Peter Robinson, EMC Corporation  
878 Scott Rotondo, Oracle  
879 Saikat Saha, SafeNet, Inc.  
880 Anil Saldhana, Red Hat  
881 Subhash Sankuratipati, NetApp  
882 Boris Schumperli, Cryptomathic  
883 Greg Singh, QuintessenceLabs  
884 David Smith, Venafi, Inc  
885 Brian Spector, Certivox  
886 Terence Spies, Voltage Security  
887 Deborah Steckroth, RouteOne LLC  
888 Michael Stevens, QuintessenceLabs  
889 Marcus Streets, Thales e-Security  
890 Satish Sundar, IBM  
891 Kiran Thota, VMware  
892 Somanchi Trinath, Freescale Semiconductor, Inc.  
893 Nathan Turajski, Thales e-Security  
894 Sean Turner, IECA, Inc.  
895 Paul Turner, Venafi, Inc.  
896 Rod Wideman, Quantum Corporation  
897 Steven Wierenga, Hewlett-Packard

898 Jin Wong, QuintessenceLabs  
899 Sameer Yami, Thales e-Security  
900 Peter Yee, EMC Corporation  
901 Krishna Yellepeddy, IBM  
902 Catherine Ying, SafeNet, Inc.  
903 Tatu Ylonen, SSH Communications Security (Tectia Corp)  
904 Michael Yoder, Vormetric. Inc.  
905 Magda Zdunkiewicz, Cryptsoft  
906 Peter Zelechowski, Election Systems & Software

## Appendix B. KMIP Specification Cross Reference

Reference Term	KMIP 1.0	KMIP 1.1	KMIP 1.2
<b>1 Introduction</b>			
<i>Non-Normative References</i>	1.3.	1.3.	1.3.
<i>Normative References</i>	1.2.	1.2.	1.2.
<i>Terminology</i>	1.1.	1.1.	1.1.
<b>2 Objects</b>			
<i>Attribute</i>	2.1.1.	2.1.1.	2.1.1.
<i>Base Objects</i>	2.1.	2.1.	2.1.
<i>Certificate</i>	2.2.1.	2.2.1.	2.2.1.
<i>Credential</i>	2.1.2.	2.1.2.	2.1.2.
<i>Data</i>	-	-	2.1.10.
<i>Data Length</i>	-	-	2.1.11.
<i>Extension Information</i>	-	2.1.9.	2.1.9.
<i>Key Block</i>	2.1.3.	2.1.3.	2.1.3.
<i>Key Value</i>	2.1.4.	2.1.4.	2.1.4.
<i>Key Wrapping Data</i>	2.1.5.	2.1.5.	2.1.5.
<i>Key Wrapping Specification</i>	2.1.6.	2.1.6.	2.1.6.
<i>MAC Data</i>	-	-	2.1.13.
<i>Managed Objects</i>	2.2.	2.2.	2.2.
<i>Nonce</i>	-	-	2.1.14.
<i>Opaque Object</i>	2.2.8.	2.2.8.	2.2.8.
<i>PGP Key</i>	-	-	2.2.9.
<i>Private Key</i>	2.2.4.	2.2.4.	2.2.4.
<i>Public Key</i>	2.2.3.	2.2.3.	2.2.3.
<i>Secret Data</i>	2.2.7.	2.2.7.	2.2.7.
<i>Signature Data</i>	-	-	2.1.12.
<i>Split Key</i>	2.2.5.	2.2.5.	2.2.5.
<i>Symmetric Key</i>	2.2.2.	2.2.2.	2.2.2.
<i>Template</i>	2.2.6.	2.2.6.	2.2.6.
<i>Template-Attribute Structures</i>	2.1.8.	2.1.8.	2.1.8.
<i>Transparent DH Private Key</i>	2.1.7.6.	2.1.7.6.	2.1.7.6.
<i>Transparent DH Public Key</i>	2.1.7.7.	2.1.7.7.	2.1.7.7.
<i>Transparent DSA Private Key</i>	2.1.7.2.	2.1.7.2.	2.1.7.2.
<i>Transparent DSA Public Key</i>	2.1.7.3.	2.1.7.3.	2.1.7.3.
<i>Transparent ECDH Private Key</i>	2.1.7.10.	2.1.7.10.	2.1.7.10.
<i>Transparent ECDH Public Key</i>	2.1.7.11.	2.1.7.11.	2.1.7.11.
<i>Transparent ECDSA Private Key</i>	2.1.7.8.	2.1.7.8.	2.1.7.8.
<i>Transparent ECDSA Public Key</i>	2.1.7.9.	2.1.7.9.	2.1.7.9.
<i>Transparent ECMQV Private Key</i>	2.1.7.12.	2.1.7.12.	2.1.7.12.
<i>Transparent ECMQV Public Key</i>	2.1.7.13.	2.1.7.13.	2.1.7.13.
<i>Transparent Key Structures</i>	2.1.7.	2.1.7.	2.1.7.
<i>Transparent RSA Private Key</i>	2.1.7.4.	2.1.7.4.	2.1.7.4.
<i>Transparent RSA Public Key</i>	2.1.7.5.	2.1.7.5.	2.1.7.5.
<i>Transparent Symmetric Key</i>	2.1.7.1.	2.1.7.1.	2.1.7.1.
<b>3 Attributes</b>			
<i>Activation Date</i>	3.19.	3.24.	3.24.
<i>Alternative Name</i>	-	-	3.40.
<i>Application Specific Information</i>	3.30.	3.36.	3.36.
<i>Archive Date</i>	3.27.	3.32.	3.32.

<b>Reference Term</b>	<b>KMIP 1.0</b>	<b>KMIP 1.1</b>	<b>KMIP 1.2</b>
<i>Attributes</i>	3	3	3
<i>Certificate Identifier</i>	3.9.	3.13.	3.13.
<i>Certificate Issuer</i>	3.11.	3.15.	3.15.
<i>Certificate Length</i>	-	3.9.	3.9.
<i>Certificate Subject</i>	3.10.	3.14.	3.14.
<i>Certificate Type</i>	3.8.	3.8.	3.8.
<i>Compromise Date</i>	3.25.	3.30.	3.30.
<i>Compromise Occurrence Date</i>	3.24.	3.29.	3.29.
<i>Contact Information</i>	3.31.	3.37.	3.37.
<i>Cryptographic Algorithm</i>	3.4.	3.4.	3.4.
<i>Cryptographic Domain Parameters</i>	3.7.	3.7.	3.7.
<i>Cryptographic Length</i>	3.5.	3.5.	3.5.
<i>Cryptographic Parameters</i>	3.6.	3.6.	3.6.
<i>Custom Attribute</i>	3.33.	3.39.	3.39.
<i>Deactivation Date</i>	3.22.	3.27.	3.27.
<i>Default Operation Policy</i>	3.13.2.	3.18.2.	3.18.2.
<i>Default Operation Policy for Certificates and Public Key Objects</i>	3.13.2.2.	3.18.2.2.	3.18.2.2.
<i>Default Operation Policy for Secret Objects</i>	3.13.2.1.	3.18.2.1.	3.18.2.1.
<i>Default Operation Policy for Template Objects</i>	3.13.2.3.	3.18.2.3.	3.18.2.3.
<i>Destroy Date</i>	3.23.	3.28.	3.28.
<i>Digest</i>	3.12.	3.17.	3.17.
<i>Digital Signature Algorithm</i>	-	3.16.	3.16.
<i>Fresh</i>	-	3.34.	3.34.
<i>Initial Date</i>	3.18.	3.23.	3.23.
<i>Key Value Location</i>	-	-	3.42.
<i>Key Value Present</i>	-	-	3.41.
<i>Last Change Date</i>	3.32.	3.38.	3.38.
<i>Lease Time</i>	3.15.	3.20.	3.20.
<i>Link</i>	3.29.	3.35.	3.35.
<i>Name</i>	3.2.	3.2.	3.2.
<i>Object Group</i>	3.28.	3.33.	3.33.
<i>Object Type</i>	3.3.	3.3.	3.3.
<i>Operation Policy Name</i>	3.13.	3.18.	3.18.
<i>Operations outside of operation policy control</i>	3.13.1.	3.18.1.	3.18.1.
<i>Original Creation Date</i>	-	-	3.43.
<i>Process Start Date</i>	3.20.	3.25.	3.25.
<i>Protect Stop Date</i>	3.21.	3.26.	3.26.
<i>Revocation Reason</i>	3.26.	3.31.	3.31.
<i>State</i>	3.17.	3.22.	3.22.
<i>Unique Identifier</i>	3.1.	3.1.	3.1.
<i>Usage Limits</i>	3.16.	3.21.	3.21.
<i>X.509 Certificate Identifier</i>	-	3.10.	3.10.
<i>X.509 Certificate Issuer</i>	-	3.12.	3.12.
<i>X.509 Certificate Subject</i>	-	3.11.	3.11.
<b>4 Client-to-Server Operations</b>			
<i>Activate</i>	4.18.	4.19.	4.19.
<i>Add Attribute</i>	4.13.	4.14.	4.14.
<i>Archive</i>	4.21.	4.22.	4.22.
<i>Cancel</i>	4.25.	4.27.	4.27.
<i>Certify</i>	4.6.	4.7.	4.7.
<i>Check</i>	4.9.	4.10.	4.10.
<i>Create</i>	4.1.	4.1.	4.1.
<i>Create Key Pair</i>	4.2.	4.2.	4.2.

<b>Reference Term</b>	<b>KMIP 1.0</b>	<b>KMIP 1.1</b>	<b>KMIP 1.2</b>
<i>Create Split Key</i>	-	-	4.38.
<i>Decrypt</i>	-	-	4.30.
<i>Delete Attribute</i>	4.15.	4.16.	4.16.
<i>Derive Key</i>	4.5.	4.6.	4.6.
<i>Destroy</i>	4.20.	4.21.	4.21.
<i>Discover Versions</i>	-	4.26.	4.26.
<i>Encrypt</i>	-	-	4.29.
<i>Get</i>	4.10.	4.11.	4.11.
<i>Get Attribute List</i>	4.12.	4.13.	4.13.
<i>Get Attributes</i>	4.11.	4.12.	4.12.
<i>Get Usage Allocation</i>	4.17.	4.18.	4.18.
<i>Hash</i>	-	-	4.37.
<i>Join Split Key</i>	-	-	4.39.
<i>Locate</i>	4.8.	4.9.	4.9.
<i>MAC</i>	-	-	4.33.
<i>MAC Verify</i>	-	-	4.34.
<i>Modify Attribute</i>	4.14.	4.15.	4.15.
<i>Obtain Lease</i>	4.16.	4.17.	4.17.
<i>Poll</i>	4.26.	4.28.	4.28.
<i>Query</i>	4.24.	4.25.	4.25.
<i>Re-certify</i>	4.7.	4.8.	4.8.
<i>Recover</i>	4.22.	4.23.	4.23.
<i>Register</i>	4.3.	4.3.	4.3.
<i>Re-key</i>	4.4.	4.4.	4.4.
<i>Re-key Key Pair</i>	-	4.5.	4.5.
<i>Revoke</i>	4.19.	4.20.	4.20.
<i>RNG Retrieve</i>	-	-	4.35.
<i>RNG Seed</i>	-	-	4.36.
<i>Sign</i>	-	-	4.31.
<i>Signature Verify</i>	-	-	4.32.
<i>Validate</i>	4.23.	4.24.	4.24.
<b>5 Server-to-Client Operations</b>			
<i>Notify</i>	5.1.	5.1.	5.1.
<i>Put</i>	5.2.	5.2.	5.2.
<b>6 Message Contents</b>			
<i>Asynchronous Correlation Value</i>	6.8.	6.8.	6.8.
<i>Asynchronous Indicator</i>	6.7.	6.7.	6.7.
<i>Attestation Capable Indicator</i>	-	-	6.17.
<i>Batch Count</i>	6.14.	6.14.	6.14.
<i>Batch Error Continuation Option</i>	6.13.	6.13.	6.13.
<i>Batch Item</i>	6.15.	6.15.	6.15.
<i>Batch Order Option</i>	6.12.	6.12.	6.12.
<i>Maximum Response Size</i>	6.3.	6.3.	6.3.
<i>Message Extension</i>	6.16.	6.16.	6.16.
<i>Operation</i>	6.2.	6.2.	6.2.
<i>Protocol Version</i>	6.1.	6.1.	6.1.
<i>Result Message</i>	6.11.	6.11.	6.11.
<i>Result Reason</i>	6.10.	6.10.	6.10.
<i>Result Status</i>	6.9.	6.9.	6.9.
<i>Time Stamp</i>	6.5.	6.5.	6.5.
<i>Unique Batch Item ID</i>	6.4.	6.4.	6.4.
<b>7 Message Format</b>			



<b>Reference Term</b>	<b>KMIP 1.0</b>	<b>KMIP 1.1</b>	<b>KMIP 1.2</b>
<i>Message Structure</i>	7.1.	7.1.	7.1.
<i>Operations</i>	7.2.	7.2.	7.2.
<b>8 Authentication</b>			
<i>Authentication</i>	8	8	8
<b>9 Message Encoding</b>			
<i>Alternative Name Type Enumeration</i>	-	-	9.1.3.2.34.
<i>Attestation Type Enumeration</i>	-	-	9.1.3.2.36.
<i>Batch Error Continuation Option Enumeration</i>	9.1.3.2.29.	9.1.3.2.30.	9.1.3.2.30.
<i>Bit Masks</i>	9.1.3.3.	9.1.3.3.	9.1.3.3.
<i>Block Cipher Mode Enumeration</i>	9.1.3.2.13.	9.1.3.2.14.	9.1.3.2.14.
<i>Cancellation Result Enumeration</i>	9.1.3.2.24.	9.1.3.2.25.	9.1.3.2.25.
<i>Certificate Request Type Enumeration</i>	9.1.3.2.21.	9.1.3.2.22.	9.1.3.2.22.
<i>Certificate Type Enumeration</i>	9.1.3.2.6.	9.1.3.2.6.	9.1.3.2.6.
<i>Credential Type Enumeration</i>	9.1.3.2.1.	9.1.3.2.1.	9.1.3.2.1.
<i>Cryptographic Algorithm Enumeration</i>	9.1.3.2.12.	9.1.3.2.13.	9.1.3.2.13.
<i>Cryptographic Usage Mask</i>	9.1.3.3.1.	9.1.3.3.1.	9.1.3.3.1.
<i>Defined Values</i>	9.1.3.	9.1.3.	9.1.3.
<i>Derivation Method Enumeration</i>	9.1.3.2.20.	9.1.3.2.21.	9.1.3.2.21.
<i>Digital Signature Algorithm Enumeration</i>	-	9.1.3.2.7.	9.1.3.2.7.
<i>Encoding Option Enumeration</i>	-	9.1.3.2.32.	9.1.3.2.32.
<i>Enumerations</i>	9.1.3.2.	9.1.3.2.	9.1.3.2.
<i>Examples</i>	9.1.2.	9.1.2.	9.1.2.
<i>Hashing Algorithm Enumeration</i>	9.1.3.2.15.	9.1.3.2.16.	9.1.3.2.16.
<i>Item Length</i>	9.1.1.3.	9.1.1.3.	9.1.1.3.
<i>Item Tag</i>	9.1.1.1.	9.1.1.1.	9.1.1.1.
<i>Item Type</i>	9.1.1.2.	9.1.1.2.	9.1.1.2.
<i>Item Value</i>	9.1.1.4.	9.1.1.4.	9.1.1.4.
<i>Key Compression Type Enumeration</i>	9.1.3.2.2.	9.1.3.2.2.	9.1.3.2.2.
<i>Key Format Type Enumeration</i>	9.1.3.2.3.	9.1.3.2.3.	9.1.3.2.3.
<i>Key Role Type Enumeration</i>	9.1.3.2.16.	9.1.3.2.17.	9.1.3.2.17.
<i>Key Value Location Type Enumeration</i>	-	-	9.1.3.2.35.
<i>Link Type Enumeration</i>	9.1.3.2.19.	9.1.3.2.20.	9.1.3.2.20.
<i>Name Type Enumeration</i>	9.1.3.2.10.	9.1.3.2.11.	9.1.3.2.11.
<i>Object Group Member Enumeration</i>	-	9.1.3.2.33.	9.1.3.2.33.
<i>Object Type Enumeration</i>	9.1.3.2.11.	9.1.3.2.12.	9.1.3.2.12.
<i>Opaque Data Type Enumeration</i>	9.1.3.2.9.	9.1.3.2.10.	9.1.3.2.10.
<i>Operation Enumeration</i>	9.1.3.2.26.	9.1.3.2.27.	9.1.3.2.27.
<i>Padding Method Enumeration</i>	9.1.3.2.14.	9.1.3.2.15.	9.1.3.2.15.
<i>Put Function Enumeration</i>	9.1.3.2.25.	9.1.3.2.26.	9.1.3.2.26.
<i>Query Function Enumeration</i>	9.1.3.2.23.	9.1.3.2.24.	9.1.3.2.24.
<i>Recommended Curve Enumeration for ECDSA, ECDH, and ECMQV</i>	9.1.3.2.5.	9.1.3.2.5.	9.1.3.2.5.
<i>Result Reason Enumeration</i>	9.1.3.2.28.	9.1.3.2.29.	9.1.3.2.29.
<i>Result Status Enumeration</i>	9.1.3.2.27.	9.1.3.2.28.	9.1.3.2.28.
<i>Revocation Reason Code Enumeration</i>	9.1.3.2.18.	9.1.3.2.19.	9.1.3.2.19.
<i>Secret Data Type Enumeration</i>	9.1.3.2.8.	9.1.3.2.9.	9.1.3.2.9.
<i>Split Key Method Enumeration</i>	9.1.3.2.7.	9.1.3.2.8.	9.1.3.2.8.
<i>State Enumeration</i>	9.1.3.2.17.	9.1.3.2.18.	9.1.3.2.18.
<i>Storage Status Mask</i>	9.1.3.3.2.	9.1.3.3.2.	9.1.3.3.2.
<i>Tags</i>	9.1.3.1.	9.1.3.1.	9.1.3.1.
<i>TTLV Encoding</i>	9.1.	9.1.	9.1.
<i>TTLV Encoding Fields</i>	9.1.1.	9.1.1.	9.1.1.
<i>Usage Limits Unit Enumeration</i>	9.1.3.2.30.	9.1.3.2.31.	9.1.3.2.31.

<b><u>Reference Term</u></b>	<b><u>KMIP 1.0</u></b>	<b><u>KMIP 1.1</u></b>	<b><u>KMIP 1.2</u></b>
<i>Validity Indicator Enumeration</i>	9.1.3.2.22.	9.1.3.2.23.	9.1.3.2.23.
<i>Wrapping Method Enumeration</i>	9.1.3.2.4.	9.1.3.2.4.	9.1.3.2.4.
<i>XML Encoding</i>	9.2.	-	-
<b>10 Transport</b>			
<i>Transport</i>	10	10	10
<b>12 KMIP Server and Client Implementation Conformance</b>			
<i>Conformance clauses for a KMIP Server</i>	12.1.	-	-
<i>KMIP Client Implementation Conformance</i>	-	12.2.	12.2.
<i>KMIP Server Implementation Conformance</i>	-	12.1.	12.1.

---

## Appendix C. Revision History

Revision	Date	Editor	Changes Made
wd01	26-June-2013	Tim Hudson	Merged version of the three committee draft documents. Updated conformance wording style. Updated test case style. Applied new OASIS template.
wd02	6-August-2013	Tim Hudson	Updated to include Permitted Test Case Variations
wd03	10-August-2013	Tim Hudson	Updated Permitted Test Case Variations
pr01update	11-June-2014	Tim Hudson	Updated following Public Review 01