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), NetApp
Saikat Saha (saikat.saha@oracle.com), Oracle

Editor:
Tim Hudson (tjh@cryptsoft.com), Cryptsoft Pty Ltd.

Related work:
This specification is related to:
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]
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
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

[KMIP-SPEC] One or more of [KMIP-SPEC-1_0], [KMIP-SPEC-1_1], [KMIP-SPEC-1_2]

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]

Candidate OASIS Standard 01. DD MMM YYYY.
1.3 Non-Normative References

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
KMIP servers that support server to client operations SHALL behave as a HTTPS client.
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.
<RequestMessage>
  <BatchItem>
    <BatchCount value="1"/>
    <MaximumResponseSize value="2048"/>
    <ProtocolVersionMajor type="Integer" value="1"/>
    <ProtocolVersionMinor type="Integer" value="0"/>
    <TimeStamp type="DateTime" value="2013-06-26T09:09:17+00:00"/>
  </BatchItem>
</ResponseMessage>

<RequestMessage>
  <BatchItem>
    <BatchCount value="1"/>
    <MaximumResponseSize value="2048"/>
    <ProtocolVersionMajor type="Integer" value="1"/>
    <ProtocolVersionMinor type="Integer" value="0"/>
    <TimeStamp type="DateTime" value="2013-06-26T09:09:17+00:00"/>
  </BatchItem>
</ResponseMessage>
```xml
<RequestPayload>
  <Operation type="Enumeration" value="Query"/>
</RequestPayload>

<ResponsePayload>
  <Operation type="Enumeration" value="QueryOperations"/>
  <Operation type="Enumeration" value="QueryObjects"/>
</ResponsePayload>

<RequestMessage>
  <BatchItem>
    <RequestPayload>
      4200780100000004820069010000002042006a020000000000000010000000
      42006b0200000040000000000000000420050020000000400000008000000000000000200000004
      0000010000000000000420050020000000400000008000000000000000200000004
      42007405000000000000000000000000000000000000000000
    </RequestPayload>
  </BatchItem>
</ResponseMessage>
```

```xml
<ResponseMessage>
  <ResponseHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1"/>
      <ProtocolVersionMinor type="Integer" value="0"/>
    </ProtocolVersion>
  </ResponseHeader>

  <BatchItem>
    <Operation type="Enumeration" value="Query"/>
    <ResultStatus type="Enumeration" value="Success"/>
    <ResponsePayload>
      <Operation type="Enumeration" value="QueryOperations"/>
      <Operation type="Enumeration" value="QueryObjects"/>
    </ResponsePayload>
  </BatchItem>
</ResponseMessage>
```
### 3.2 Mandatory HTTPS Profile Test Cases KMIP v1.1

#### 3.2.1 MSGENC-HTTPS-M-1-11 - 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
00000100: 00 00 00 1a 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 19 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 09 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 11 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 02 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 04 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 15 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 16 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 00 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 00 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 07 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 1b 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 1c 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 00 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 00 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 10 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 00 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 00 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 00 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 00 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 00 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 00 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 00 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 00 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 00 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 00 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....
00000100: 00 00 00 00 00 00 00 00-42 00 5c 05 00 00 00 04 .......B.\.....

<RequestMessage>
    <RequestHeader>
        <ProtocolVersion>
            <ProtocolVersionMajor  type="Integer" value="1"/>
            <ProtocolVersionMinor  type="Integer" value="1"/>
        </ProtocolVersion>
        <MaximumResponseSize type="Integer" value="256"/>
    </RequestHeader>
    <Operation type="Enumeration" value="Query"/>
    <RequestPayload>
        <QueryFunction type="Enumeration" value="QueryOperations"/>
        <QueryFunction type="Enumeration" value="QueryObjects"/>
    </RequestPayload>
</RequestMessage>
```

null
<ResponseMessage>
  <ResponseHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1"/>
      <ProtocolVersionMinor type="Integer" value="1"/>
    </ProtocolVersion>
    <TimeStamp type="DateTime" value="2014-06-10T08:03:34+00:00"/>
    <BatchCount type="Integer" value="1"/>
  </ResponseHeader>
  <BatchItem>
    <Operation type="Enumeration" value="Query"/>
    <Operation type="Enumeration" value="Success"/>
    <ResponsePayload>
      <Operation type="Enumeration" value="Query"/>
      <Operation type="Enumeration" value="Locate"/>
      <Operation type="Enumeration" value="Destroy"/>
      <Operation type="Enumeration" value="Get"/>
      <Operation type="Enumeration" value="Create"/>
      <Operation type="Enumeration" value="Register"/>
      <Operation type="Enumeration" value="GetAttributes"/>
      <Operation type="Enumeration" value="ModifyAttributeList"/>
      <Operation type="Enumeration" value="AddAttribute"/>
      <Operation type="Enumeration" value="ModifyAttribute"/>
      <Operation type="Enumeration" value="DeleteAttribute"/>
      <Operation type="Enumeration" value="Activate"/>
      <Operation type="Enumeration" value="Revoke"/>
      <Operation type="Enumeration" value="Poll"/>
      <Operation type="Enumeration" value="Cancel"/>
    </ResponsePayload>
    <BatchItem>
      <Operation type="Enumeration" value="Check"/>
    </BatchItem>
  </BatchItem>
</ResponseMessage>
<Operation type="Enumeration" value="GetUsageAllocation"/>
<Operation type="Enumeration" value="CreateKeyPair"/>
<Operation type="Enumeration" value="ReKey"/>
<Operation type="Enumeration" value="Archive"/>
<Operation type="Enumeration" value="Recover"/>
<Operation type="Enumeration" value="ObtainLease"/>
<Operation type="Enumeration" value="ReKeyKeyPair"/>
<Operation type="Enumeration" value="Certify"/>
<Operation type="Enumeration" value="ReCertify"/>
<Operation type="Enumeration" value="DiscoverVersions"/>
<Operation type="Enumeration" value="Notify"/>
<Operation type="Enumeration" value="Put"/>
<ObjectType type="Enumeration" value="Certificate"/>
<ObjectType type="Enumeration" value="SymmetricKey"/>
<ObjectType type="Enumeration" value="SecretData"/>
<ObjectType type="Enumeration" value="Public"/>
<ObjectType type="Enumeration" value="PrivateKey"/>
<ObjectType type="Enumeration" value="Template"/>
<ObjectType type="Enumeration" value="OpaqueObject"/>
<ObjectType type="Enumeration" value="SplitKey"/>
</ResponsePayload>
</BatchItem>
</ResponseMessage>

42007b010000002c042007a0100000048420069010000002042006a02000000040000000100000000
42006b020000000040000000010000000042009209000000008000000005396bc4242000d0200000004
00000001000000042000f01000026842005c05000000040000001800000000420070f0500000004
0000000000000000042007c0100000024042005c050000000400000018000000004205c0500000004
0000000000000000042002c050000000400000014000000004205c05000000040000000a00000000
4200c0500000004000000010000000004205c050000000400000003000000004205c0500000004
000000b00000004205c05000000040000000c000000004205c05000000040000000d00000000
4205c05000000040000000e000000004205c05000000040000000f000000004205c0500000004
0000001200000004205c050000000400000013000000004205c050000000400000001a00000000
4205c05000000040000000400000009000000090000000900000009000000090000000900000009
4205c0500000004000000040000000f000000004205c0500000004000000040000000400000004
0000000100000004205c050000000400000001c000000004205c05000000040000000000000000
4205c05000000040000000d000000004205c05000000040000000d000000004205c0500000004
0000001b00000004205c050000000400000001c000000004205c05000000040000000000000000
4205c050000000400000006000000004205c050000000400000006000000004205c0500000004
0000000000: 48 54 54 0d 30 0d 0a 0d 0a .Content-Type: a
000000010: 0a 3a 20 73 75 6c 6c 69 6e 67 2c 2f 20 22 61 70 70 6c 69 63 65 2f 6f 72 6f 6a
.Http/1.1
000000020: 20 46 6f 72 6f 20 31 2e 31 20 20 20 20 ...
Application/octet-
000000030: 2d 73 74 65 6e 74 65 72 2e 70 72 6f 74 68 65 20 31 2e 31 20 20 20 20 ...
/timestamp
000000040: 2d 73 74 65 6e 74 65 72 2e 70 72 6f 74 68 65 20 31 2e 31 20 20 20 20 ...
/timestamp
000000050: 42 00 7b 01 00 00 02 c0-42 00 7a 01 00 00 00 00 B. {...
000000060: 42 00 69 01 00 00 00 20-42 00 6a 02 00 00 00 00 B.i.... B.j....
000000070: 42 00 00 00 00 00 00 00 B.k..... B.l.....
000000080: 42 00 00 00 00 00 00 00 B.m..... B.n.....
000000090: 42 00 00 00 00 00 00 00 B.o..... B.p.....
0000000a0: 42 00 00 00 00 00 00 00 B.q..... B.r.....
0000000b0: 42 00 00 00 00 00 00 00 B.s..... B.t.....
0000000c0: 42 00 00 00 00 00 00 00 B.u..... B.v.....
0000000d0: 42 00 00 00 00 00 00 00 B.w..... B.x.....
0000000e0: 42 00 00 00 00 00 00 00 B.y..... B.z.....
0000000f0: 42 00 00 00 00 00 00 00 B.\..... B.\.....
000000100: 42 00 00 00 00 00 00 00 B.\\..... B.\\.....
000000110: 42 00 00 00 00 00 00 00 B.^..... B.^.....
000000120: 42 00 00 00 00 00 00 00 B._..... B._.....
000000130: 42 00 00 00 00 00 00 00 B.\..... B.\.....
000000140: 42 00 00 00 00 00 00 00 B.\..... B.\.....
000000150: 42 00 00 00 00 00 00 00 B.\..... B.\.....
000000160: 42 00 00 00 00 00 00 00 B.\..... B.\.....
### 3.3 Mandatory HTTPS Profile Test Cases KMIP v1.2

#### 3.3.1 MSGENC-HTTPS-M-1-12 - 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.

```xml
# TIME 0
<RequestMessage>
  <RequestHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor>1</ProtocolVersionMajor>
    </ProtocolVersion>
    <ProtocolVersionMinor>2</ProtocolVersionMinor>
  </RequestHeader>
  <BatchCount>256</BatchCount>
  <MaximumResponseSize>1</MaximumResponseSize>
  <QueryFunction type="QueryOperations"/>
  <QueryFunction type="QueryObjects"/>
  <Operation type="Query"/>
</RequestMessage>
```

```xml
42007801000000094202007701000000094200690100000002042006a0200000000400000000001000000
42006b020000000400000002000000040200500200000004000000010000000402000002000000040200000200000004
0000001000000004200000000000003842005c05000000040000001800000004200790100000020
420074050000000400000001000000042007405000000040000000200000000
```
<RequestMessage>
  <RequestHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1"/>
      <ProtocolVersionMinor type="Integer" value="2"/>
    </ProtocolVersion>
  </RequestHeader>
  <Operation type="Enumeration" value="Query"/>
  <ResultStatus type="Enumeration" value="OperationFailed"/>
  <ResultReason type="Enumeration" value="ResponseTooLarge"/>
  <ResultMessage type="TextString" value="TOO_LARGE"/>
</RequestMessage>

<RequestMessage>
  <RequestHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1"/>
      <ProtocolVersionMinor type="Integer" value="2"/>
    </ProtocolVersion>
  </RequestHeader>
  <Operation type="Enumeration" value="Query"/>
  <ResultStatus type="Enumeration" value="OperationFailed"/>
  <ResultReason type="Enumeration" value="ResponseTooLarge"/>
  <ResultMessage type="TextString" value="TOO_LARGE"/>
</RequestMessage>

<RequestMessage>
  <RequestHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1"/>
      <ProtocolVersionMinor type="Integer" value="2"/>
    </ProtocolVersion>
  </RequestHeader>
  <Operation type="Enumeration" value="Query"/>
  <ResultStatus type="Enumeration" value="OperationFailed"/>
  <ResultReason type="Enumeration" value="ResponseTooLarge"/>
  <ResultMessage type="TextString" value="TOO_LARGE"/>
</RequestMessage>
null
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
# 1. Replace brackets with space
noBrackets = re.sub('[()]', ' ', enumName)
# 2. replace \W with space if followed by letter, lower
nonWordToSpace = re.sub('\\W([A-Za-z][a-z])', r' \1', noBrackets)
# 3. non-word to underscore
words = [re.sub('\\W', '_ ', s) for s in nonWordToSpace.split()]
# 4. move numbers to end of first word
words[0] = re.sub('^\((d+)\.*\)', r'\2\1', words[0])
# 5. capitalize first letter of each word
words = [re.sub('^.', s[0].upper(), s) for s in words]
# 6. concatenate
enumNameCamel = ''.join(words)

Example python name normalization code

# 1. Replace brackets with space
$enumName=~s/\[()]/ /g;
# 2. replace \W with space if followed by letter, lower
$enumName=~s/\\W([A-Za-z][a-z])/ /g;
# 3. non-word to underscore
@words=split(/ /,$enumName);
for($i=0;$i<=$#words;$i++) { $words[$i]=~s/\\W/_/g; }
# 4. move numbers to end of first word
$words[0] =~ s/^\((d+)\.*\)/\2\1/;
# 5. capitalize first letter of each word
for($i=0;$i<=$#words;$i++) { substr($words[$i],0,1)=~tr/a-z/A-Z/;
}
# 6. concatenate
$enumNameCamel = join('',@words);

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’.

```json
{"tag": "ActivationDate", "type": "DateTime", "value": "2001-01-01T10:00:00+10:00"}
{"tag": "0x420001", "type": "ByteString", "value": "a1b2c3d4"}
{"tag": "PrivateTemplateAttribute", "type": "Structure", "value": []}
{"tag": "0x545352", "type": "TextString", "value": "This is an extension"}
{"tag": "WELL_KNOWN_EXTENSION", "type": "TextString", "value": "This is an extension"}
```

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.

```json
{"tag": "0x420001", "type": "DateTime", "value": "2001-01-01T10:00:00+10:00"}
{"tag": "IVCounterNonce", "type": "ByteString", "value": "a1b2c3d4"}
{"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.

```json
{"tag": "ProtocolVersion", "type": "Structure", "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.

```json
{"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.

```json
{"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.

```json
{  "tag": "0x540001",  "type":"LongInteger",  "value":"0xfffffffffffffffff"}
{  "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.

```json
{  "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

```json
{  "tag": "0x420057",  "type":"Enumeration",  "value":2}
{  "tag": "0x000000003a505520",  "type":"Enumeration",  "value":"0x00000000"}
```

4.1.6.8 Boolean
For JSON, value must be either a hex string, or a JSON Boolean ‘true’ or ‘false’.

```json
{  "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

```json
{  "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.

```json
{  "tag": "MACSignature",  "type":"ByteString",  "value":"C5OF77"}
```

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:

```text
'-? yyyy '-' mm '-' dd 'T' hh ':' 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).

```json
{  "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.

```json
{  "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.

```xml
# 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>

42007801000000094200770100000048420069010000002042006a02000000040000001000000000
42006b02000000400000000000000004200500200000004000000100000000004200002000000004
0000000100000000042000f01000003842005c05000000400000001800000004200790100000020
42007405000000400000000100000000420074050000000040000000200000000

{"tag":"RequestMessage", "value":[
{"tag":"RequestHeader", "value":[
{"tag":"ProtocolVersion", "value":[
{"tag":"ProtocolVersionMajor", "type":"Integer", "value":"0x00000001"},
{"tag":"ProtocolVersionMinor", "type":"Integer", "value":"0x00000000"}

"tag":"RequestMessage", "value":[
{"tag":"RequestHeader", "value":[
{"tag":"ProtocolVersion", "value":[
{"tag":"ProtocolVersionMajor", "type":"Integer", "value":"0x00000001"},
{"tag":"ProtocolVersionMinor", "type":"Integer", "value":"0x00000000"}
```

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>
<RequestPayload>
  <QueryFunction type="Enumeration" value="QueryOperations"/>
  <QueryFunction type="Enumeration" value="QueryObjects"/>
</RequestPayload>

</BatchItem>
</RequestMessage>

{"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"}
    ]}
  ]}
]}

<ResponseMessage>
  <ResponseHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1"/>
      <ProtocolVersionMinor type="Integer" value="0"/>
    </ProtocolVersion>
</ResponseHeader>

  <BatchCount type="Integer" value="1"/>
</ResponseHeader>

  <BatchItem>
    <Operation type="Enumeration" value="Query"/>
    <ResultStatus type="Enumeration" value="Success"/>
  </BatchItem>
  <ResponsePayload>
    <Operation type="Enumeration" value="Query"/>
    <Operation type="Enumeration" value="Locate"/>
    <Operation type="Enumeration" value="Destroy"/>
    <Operation type="Enumeration" value="Get"/>
    <Operation type="Enumeration" value="Create"/>
    <Operation type="Enumeration" value="Register"/>
    <Operation type="Enumeration" value="GetAttributes"/>
    <Operation type="Enumeration" value="GetAttributeList"/>
    <Operation type="Enumeration" value="AddAttribute"/>
    <Operation type="Enumeration" value="ModifyAttribute"/>
    <Operation type="Enumeration" value="DeleteAttribute"/>
    <Operation type="Enumeration" value="Activate"/>
    <Operation type="Enumeration" value="Revoke"/>
    <Operation type="Enumeration" value="Poll"/>
    <Operation type="Enumeration" value="Cancel"/>
    <Operation type="Enumeration" value="Check"/>
    <Operation type="Enumeration" value="GetUsageAllocation"/>
    <Operation type="Enumeration" value="CreateKeyPair"/>
  </ResponsePayload>

kmip-addtl-msg-enc-v1.0-csd02
Standards Track Work Product Copyright © OASIS Open 2014. All Rights Reserved.

19 June 2014
Page 29 of 67
<Operation type="Enumeration" value="ReKey"/>
<Operation type="Enumeration" value="Archive"/>
<Operation type="Enumeration" value="Recover"/>
<Operation type="Enumeration" value="ObtainLease"/>
<Operation type="Enumeration" value="Certify"/>
<Operation type="Enumeration" value="ReCertify"/>
<Operation type="Enumeration" value="Notify"/>
<Operation type="Enumeration" value="Put"/>
<ObjectType type="Enumeration" value="Certificate"/>
<ObjectType type="Enumeration" value="SymmetricKey"/>
<ObjectType type="Enumeration" value="SecretData"/>
<ObjectType type="Enumeration" value="PublicKey"/>
<ObjectType type="Enumeration" value="PrivateKey"/>
<ObjectType type="Enumeration" value="Template"/>
<ObjectType type="Enumeration" value="OpaqueObject"/>
<ObjectType type="Enumeration" value="SplitKey"/>
</ResponsePayload>
</BatchItem>
</ResponseMessage>
<table>
<thead>
<tr>
<th>tag</th>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>Enumeration</td>
<td>GetUsageAllocation</td>
</tr>
<tr>
<td>Operation</td>
<td>Enumeration</td>
<td>CreateKeyPair</td>
</tr>
<tr>
<td>Operation</td>
<td>Enumeration</td>
<td>Encrypt</td>
</tr>
<tr>
<td>Operation</td>
<td>Enumeration</td>
<td>Archive</td>
</tr>
<tr>
<td>Operation</td>
<td>Enumeration</td>
<td>ReKey</td>
</tr>
<tr>
<td>Operation</td>
<td>Enumeration</td>
<td>DestroyKey</td>
</tr>
<tr>
<td>Operation</td>
<td>Enumeration</td>
<td>Recover</td>
</tr>
<tr>
<td>Operation</td>
<td>Enumeration</td>
<td>ObtainLease</td>
</tr>
<tr>
<td>Operation</td>
<td>Enumeration</td>
<td>Notify</td>
</tr>
<tr>
<td>ObjectType</td>
<td>Enumeration</td>
<td>Certificate</td>
</tr>
<tr>
<td>ObjectType</td>
<td>Enumeration</td>
<td>SymmetricKey</td>
</tr>
<tr>
<td>ObjectType</td>
<td>Enumeration</td>
<td>SecretData</td>
</tr>
<tr>
<td>ObjectType</td>
<td>Enumeration</td>
<td>PublicKey</td>
</tr>
<tr>
<td>ObjectType</td>
<td>Enumeration</td>
<td>PrivateKey</td>
</tr>
<tr>
<td>ObjectType</td>
<td>Enumeration</td>
<td>Template</td>
</tr>
<tr>
<td>ObjectType</td>
<td>Enumeration</td>
<td>OpaqueObject</td>
</tr>
<tr>
<td>ObjectType</td>
<td>Enumeration</td>
<td>SplitKey</td>
</tr>
</tbody>
</table>

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

#### 5.2.1 MSGENC-JSON-M-1-11 - 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="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>
```

```json
{"tag":"RequestMessage", "value":[]}
{"tag":"RequestHeader", "value":[]}
{"tag":"ProtocolVersion", "value":[]}
{"tag":"ProtocolVersionMajor", "type":"Integer", "value":"0x00000001"},
{"tag":"ProtocolVersionMinor", "type":"Integer", "value":"0x00000001"}
```
<ResponseMessage>
  <ResponseHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1"/>
      <ProtocolVersionMinor type="Integer" value="1"/>
    </ProtocolVersion>
    <MaximumResponseSize type="Integer" value="2048"/>
    <BatchCount type="Integer" value="0x00000001"/>
  </ResponseHeader>
  <BatchItem>
    <Operation type="Enumeration" value="Query"/>
    <RequestPayload>
      <!-- Request payload content here -->
    </RequestPayload>
  </BatchItem>
</ResponseMessage>

<RequestMessage>
  <RequestHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1"/>
      <ProtocolVersionMinor type="Integer" value="1"/>
    </ProtocolVersion>
    <MaximumResponseSize type="Integer" value="0x00000100"/>
    <BatchCount type="Integer" value="0x00000001"/>
  </RequestHeader>
  <BatchItem>
    <Operation type="Enumeration" value="Query"/>
    <RequestPayload>
      <!-- Request payload content here -->
    </RequestPayload>
  </BatchItem>
</RequestMessage>
<Operation type="Enumeration" value="Recover"/>
<Operation type="Enumeration" value="ObtainLease"/>
<Operation type="Enumeration" value="ReKeyKeyPair"/>
<Operation type="Enumeration" value="Certify"/>
<Operation type="Enumeration" value="ReCertify"/>
<Operation type="Enumeration" value="DiscoverVersions"/>
<Operation type="Enumeration" value="Notify"/>
<Operation type="Enumeration" value="Put"/>
<ObjectType type="Enumeration" value="Certificate"/>
<ObjectType type="Enumeration" value="SymmetricKey"/>
<ObjectType type="Enumeration" value="PublicKey"/>
<ObjectType type="Enumeration" value="PrivateKey"/>
<ObjectType type="Enumeration" value="Template"/>
<ObjectType type="Enumeration" value="OpaqueObject"/>
<ObjectType type="Enumeration" value="SecretData"/>
<ObjectType type="Enumeration" value="SymmetricKey"/>
<ObjectType type="Enumeration" value="Certificate"/>
<ObjectType type="Enumeration" value="Template"/>
<ObjectType type="Enumeration" value="OpaqueObject"/>
<ObjectType type="Enumeration" value="DiscoverVersions"/>
<ObjectType type="Enumeration" value="GetAttributes"/>
<ObjectType type="Enumeration" value="DeleteAttribute"/>
<ObjectType type="Enumeration" value="Activate"/>
<ObjectType type="Enumeration" value="Revoke"/>
<ObjectType type="Enumeration" value="Register"/>
<ObjectType type="Enumeration" value="Locate"/>
<ObjectType type="Enumeration" value="Get"/>
<ObjectType type="Enumeration" value="Create"/>
<ObjectType type="Enumeration" value="Register"/>
<ObjectType type="Enumeration" value="GetAttributes"/>
<ObjectType type="Enumeration" value="GetAttributeList"/>
<ObjectType type="Enumeration" value="AddAttribute"/>
<ObjectType type="Enumeration" value="ModifyAttribute"/>
<ObjectType type="Enumeration" value="ObtainLease"/>
<ObjectType type="Enumeration" value="ReKeyKeyPair"/>
<ObjectType type="Enumeration" value="Certify"/>
<ObjectType type="Enumeration" value="ReCertify"/>
<ObjectType type="Enumeration" value="DiscoverVersions"/>
<ObjectType type="Enumeration" value="Notify"/>
<ObjectType type="Enumeration" value="Put"/>
<ObjectType type="Enumeration" value="Certificate"/>
<ObjectType type="Enumeration" value="SymmetricKey"/>
<ObjectType type="Enumeration" value="PublicKey"/>
<ObjectType type="Enumeration" value="PrivateKey"/>
<ObjectType type="Enumeration" value="Template"/>
<ObjectType type="Enumeration" value="OpaqueObject"/>
<ObjectType type="Enumeration" value="SecretData"/>
<ObjectType type="Enumeration" value="SymmetricKey"/>
<ObjectType type="Enumeration" value="Certificate"/>
<ObjectType type="Enumeration" value="Template"/>
<ObjectType type="Enumeration" value="OpaqueObject"/>
<ObjectType type="Enumeration" value="DiscoverVersions"/>
<ObjectType type="Enumeration" value="GetAttributes"/>
<ObjectType type="Enumeration" value="DeleteAttribute"/>
<ObjectType type="Enumeration" value="Activate"/>
<ObjectType type="Enumeration" value="Revoke"/>
<ObjectType type="Enumeration" value="Register"/>
<ObjectType type="Enumeration" value="Locate"/>
<ObjectType type="Enumeration" value="Get"/>
<ObjectType type="Enumeration" value="Create"/>
5.3 Mandatory JSON Profile Test Cases KMIP v1.2

5.3.1 MSGENC-JSON-M-1-12 - 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="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>
```

<Operation type="Enumeration" value="Query"/>

<RequestPayload>
  <QueryFunction type="Enumeration" value="QueryOperations"/>
  <QueryFunction type="Enumeration" value="QueryObjects"/>
</RequestPayload>

</BatchItem>

</RequestMessage>

{"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"}
            ]}
        ]}
    ]}
  ]}
}<ResponseMessage>

  <ResponseHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1"/>
      <ProtocolVersionMinor type="Integer" value="2"/>
    </ProtocolVersion>
  </ResponseHeader>

  <BatchItem>
    <Operation type="Enumeration" value="Query"/>
    <ResultStatus type="Enumeration" value="Success"/>
    <ResponsePayload>
      <Operation type="Enumeration" value="Query"/>
      <Operation type="Enumeration" value="Locate"/>
      <Operation type="Enumeration" value="Destroy"/>
      <Operation type="Enumeration" value="Get"/>
      <Operation type="Enumeration" value="Create"/>
      <Operation type="Enumeration" value="Register"/>
      <Operation type="Enumeration" value="GetAttributes"/>
      <Operation type="Enumeration" value="GetAttributeList"/>
      <Operation type="Enumeration" value="AddAttribute"/>
      <Operation type="Enumeration" value="ModifyAttribute"/>
      <Operation type="Enumeration" value="DeleteAttribute"/>
      <Operation type="Enumeration" value="Activate"/>
      <Operation type="Enumeration" value="Revoke"/>
      <Operation type="Enumeration" value="Poll"/>
      <Operation type="Enumeration" value="Cancel"/>
      <Operation type="Enumeration" value="Check"/>
      <Operation type="Enumeration" value="GetUsageAllocation"/>
      <Operation type="Enumeration" value="CreateKeyPair"/>
    </ResponsePayload>
  </BatchItem>

</ResponseMessage>
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="Hash"/>
0099  <Operation type="Enumeration" value="CreateSplitKey"/>
0100  <Operation type="Enumeration" value="JoinSplitKey"/>
0101  <ObjectType type="Enumeration" value="Certificate"/>
0102  <ObjectType type="Enumeration" value="SymmetricKey"/>
0103  <ObjectType type="Enumeration" value="SecretData"/>
0104  <ObjectType type="Enumeration" value="PublicKey"/>
0105  <ObjectType type="Enumeration" value="PrivateKey"/>
0106  <ObjectType type="Enumeration" value="Template"/>
0107  <ObjectType type="Enumeration" value="OpaqueObject"/>
0108  <ObjectType type="Enumeration" value="SplitKey"/>
0109  <ObjectType type="Enumeration" value="PGPKey"/>
0110  </ResponsePayload>
0111  </BatchItem>
0112  </ResponseMessage>
0113

42007b010000038042007a01000004842006901000002042006a020000004000000010000000
42006b0200000004000000020000000420092909000000080000000005396b0c042000d0200000004
000000010000000500000000400000001800000004200270f0500000004
000000000000042007c0100003042005c050000000400000018000000042005c0500000004
000000000000042005c050000000400000014000000042005c0500000004000000000a00000000
000000012000000042005c050000000400000013000000042005c050000000400000001a00000000
000000000000042005c050000000400000010000000042005c05000000040000000300000000
000000000000042005c0500000004000000020000000042005c05000000040000000100000000
000000000000042005c0500000004000000019000000042005c05000000040000000900000000
000000000000042005c0500000004000000020000000042005c05000000040000000400000000
42005c050000000400000015000000042005c050000000400000016000000042005c0500000004
00000001000000042005c05000000040000001d000000042005c05000000040000000600000000
00000001000000042005c050000000400000010000000042005c050000000400000001e00000000
00000001b000000042005c05000000040000001c000000042005c0500000004000000025000000
000000000000042005c0500000004000000026000000042005c0500000004000000001f00000000
000000000000042005c0500000004000000021000000042005c0500000004000000022000000
000000000000042005c0500000004000000023000000042005c05000000040000000240000000
000000000000042005c0500000004000000028000000042005c05000000040000000290000000
000000000000042005c0500000004000000010000000042005c05000000040000000200000000
000000000000042005c0500000004000000023000000042005c05000000040000000240000000
000000000000042005c0500000004000000028000000042005c05000000040000000290000000
42005705000000040000000100000000420057050000000400000002000000004200570500000004
00000000000200000420057050000000400000003000000004200570500000004000000040000000
420057050000000400000003000000004200570500000004000000040000000040000000
000000050000042005705000000040000000900000000

{"tag":"ResponseMessage", "value":[]
{"tag":"ResponseHeader", "value":
{"tag":"ProtocolVersion", "value":[]
{"tag":"ObjectType", "value":[]
{"tag":"ObjectType", "value":[]
{"tag":"ObjectType", "value":[]
{}}}]}
}
343

```json
{
  "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": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "Locate"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "Get"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "Create"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "GetAttributes"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "GetAttributeList"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "AddAttribute"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "ModifyAttribute"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "DeleteAttribute"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "Activate"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "Revoke"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "Poll"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "Cancel"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "Check"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "GetUsageAllocation"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "CreateKeyPair"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "ReKey"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "Archive"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "Recover"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "ObtainLease"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "ReKeyKeyPair"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "Certify"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "ReCertify"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "DiscoverVersions"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "Notify"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "Put"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "RNGRetrieve"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "RNGSeed"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "Encrypt"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "Decrypt"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "Sign"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "SignatureVerify"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "MAC"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "MACVerify"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "Hash"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "CreateSplitKey"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "Operation",
    "type": "Enumeration",
    "value": "JoinSplitKey"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "ObjectType",
    "type": "Enumeration",
    "value": "Certificate"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "ObjectType",
    "type": "Enumeration",
    "value": "SymmetricKey"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "ObjectType",
    "type": "Enumeration",
    "value": "SecretData"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "ObjectType",
    "type": "Enumeration",
    "value": "PublicKey"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "ObjectType",
    "type": "Enumeration",
    "value": "PrivateKey"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "ObjectType",
    "type": "Enumeration",
    "value": "Template"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "ObjectType",
    "type": "Enumeration",
    "value": "OpaqueObject"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "ObjectType",
    "type": "Enumeration",
    "value": "SplitKey"
  },
  "tag": "BatchItem",
  "value": {
    "tag": "ObjectType",
    "type": "Enumeration",
    "value": "PGPKey"
  }
}
```
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
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.

```xml
<TTLV tag="0x420001" name="ActivationDate" 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.

```xml
<ActivationDate xmlns="urn:oasis:tc:kmip:xmlns" type="DateTime" value="2001-01-01T10:00:00+10:00"/>
```

6.1.6.2 Structure

For XML, sub-items are nested elements.

```xml
<ProtocolVersion type="Structure">
    <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

```xml
<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.

```xml
<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

```xml
<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

```xml
<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.

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

### 6.1.6.8 Boolean

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

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

### 6.1.6.9 Text String

XML uses [XML-SCHEMA] type xsd:string

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

### 6.1.6.10 Byte String

XML uses [XML-SCHEMA] type xsd:hexBinary

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

### 6.1.6.11 Date-Time

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

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

### 6.1.6.12 Interval

XML uses [XML-SCHEMA] type xsd:unsignedInt

```xml
<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.

```xml
# 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>
42007801000000904200770100000048420069010000002042006a0200000004000000010000000042006b0200000004
000000010000000042000f010000003842005c050000000400000018000000004200790100000020
4200740500000004000000100000000420074050000000400000020000000000
0018  <ResponseMessage>
0019    <ResponseHeader>
0020      <ProtocolVersion>
0021        <ProtocolVersionMajor type="Integer" value="1"/>
```
### 7.2 Mandatory XML Profile Test Cases KMIP v1.1

#### 7.2.1 MSGENC-XML-M-1-11 - 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.

```xml
# 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>

# TIME 1
0018 <ResponseMessage>
0019   <ResponseHeader>
0020     <ProtocolVersion>
0021       <ProtocolVersionMajor type="Integer" value="1"/>
0022       <ProtocolVersionMinor type="Integer" value="1"/>
0023     </ProtocolVersion>
0024     <TimeStamp type="DateTime" value="2014-06-10T08:03:34+00:00"/>
0025     <BatchCount type="Integer" value="1"/>
0026   </ResponseHeader>
0027 <BatchItem>
0028   <Operation type="Enumeration" value="Query"/>
0029   <ResultStatus type="Enumeration" value="OperationFailed"/>
0030   <ResultReason type="Enumeration" value="ResponseTooLarge"/>
0031   <ResultMessage type="TextString" value="TOO_LARGE"/>
0032 </BatchItem>
0033 </ResponseMessage>
```
<table>
<thead>
<tr>
<th>Line</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>0042</td>
<td><code>&lt;BatchItem&gt;</code></td>
</tr>
<tr>
<td>0043</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Query&quot;/&gt;</code></td>
</tr>
<tr>
<td>0044</td>
<td><code>&lt;RequestPayload&gt;</code></td>
</tr>
<tr>
<td>0045</td>
<td><code>&lt;QueryFunction type=&quot;Enumeration&quot; value=&quot;QueryOperations&quot;/&gt;</code></td>
</tr>
<tr>
<td>0046</td>
<td><code>&lt;QueryFunction type=&quot;Enumeration&quot; value=&quot;QueryObjects&quot;/&gt;</code></td>
</tr>
<tr>
<td>0047</td>
<td><code>&lt;/RequestPayload&gt;</code></td>
</tr>
<tr>
<td>0048</td>
<td><code>&lt;/BatchItem&gt;</code></td>
</tr>
<tr>
<td>0049</td>
<td><code>&lt;/RequestMessage&gt;</code></td>
</tr>
<tr>
<td>0050</td>
<td><code>&lt;ResponseMessage&gt;</code></td>
</tr>
<tr>
<td>0051</td>
<td><code>&lt;ResponseHeader&gt;</code></td>
</tr>
<tr>
<td>0052</td>
<td><code>&lt;ProtocolVersion&gt;</code></td>
</tr>
<tr>
<td>0053</td>
<td><code>&lt;ProtocolVersionMajor type=&quot;Integer&quot; value=&quot;1&quot;/&gt;</code></td>
</tr>
<tr>
<td>0054</td>
<td><code>&lt;ProtocolVersionMinor type=&quot;Integer&quot; value=&quot;1&quot;/&gt;</code></td>
</tr>
<tr>
<td>0055</td>
<td><code>&lt;/ProtocolVersion&gt;</code></td>
</tr>
<tr>
<td>0056</td>
<td><code>&lt;TimeStamp type=&quot;DateTime&quot; value=&quot;2014-06-10T08:03:34+00:00&quot;/&gt;</code></td>
</tr>
<tr>
<td>0057</td>
<td><code>&lt;BatchCount type=&quot;Integer&quot; value=&quot;1&quot;/&gt;</code></td>
</tr>
<tr>
<td>0058</td>
<td><code>&lt;/ResponseHeader&gt;</code></td>
</tr>
<tr>
<td>0059</td>
<td><code>&lt;BatchItem&gt;</code></td>
</tr>
<tr>
<td>0060</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Query&quot;/&gt;</code></td>
</tr>
<tr>
<td>0061</td>
<td><code>&lt;ResultStatus type=&quot;Enumeration&quot; value=&quot;Success&quot;/&gt;</code></td>
</tr>
<tr>
<td>0062</td>
<td><code>&lt;ResponsePayload&gt;</code></td>
</tr>
<tr>
<td>0063</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Query&quot;/&gt;</code></td>
</tr>
<tr>
<td>0064</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Locate&quot;/&gt;</code></td>
</tr>
<tr>
<td>0065</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Destroy&quot;/&gt;</code></td>
</tr>
<tr>
<td>0066</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Get&quot;/&gt;</code></td>
</tr>
<tr>
<td>0067</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Create&quot;/&gt;</code></td>
</tr>
<tr>
<td>0068</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Register&quot;/&gt;</code></td>
</tr>
<tr>
<td>0069</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;GetAttributes&quot;/&gt;</code></td>
</tr>
<tr>
<td>0070</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;GetAttributeList&quot;/&gt;</code></td>
</tr>
<tr>
<td>0071</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;AddAttribute&quot;/&gt;</code></td>
</tr>
<tr>
<td>0072</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;ModifyAttribute&quot;/&gt;</code></td>
</tr>
<tr>
<td>0073</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;DeleteAttribute&quot;/&gt;</code></td>
</tr>
<tr>
<td>0074</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Activate&quot;/&gt;</code></td>
</tr>
<tr>
<td>0075</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Revoke&quot;/&gt;</code></td>
</tr>
<tr>
<td>0076</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Poll&quot;/&gt;</code></td>
</tr>
<tr>
<td>0077</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Cancel&quot;/&gt;</code></td>
</tr>
<tr>
<td>0078</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Check&quot;/&gt;</code></td>
</tr>
<tr>
<td>0079</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;GetUsageAllocation&quot;/&gt;</code></td>
</tr>
<tr>
<td>0080</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;CreateKeyPair&quot;/&gt;</code></td>
</tr>
<tr>
<td>0081</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;ReKey&quot;/&gt;</code></td>
</tr>
<tr>
<td>0082</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Archive&quot;/&gt;</code></td>
</tr>
<tr>
<td>0083</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Recover&quot;/&gt;</code></td>
</tr>
<tr>
<td>0084</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;ObtainLease&quot;/&gt;</code></td>
</tr>
<tr>
<td>0085</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;ReKeyKeyPair&quot;/&gt;</code></td>
</tr>
<tr>
<td>0086</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Certify&quot;/&gt;</code></td>
</tr>
<tr>
<td>0087</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;ReCertify&quot;/&gt;</code></td>
</tr>
<tr>
<td>0088</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;DiscoverVersions&quot;/&gt;</code></td>
</tr>
<tr>
<td>0089</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Notify&quot;/&gt;</code></td>
</tr>
<tr>
<td>0090</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Put&quot;/&gt;</code></td>
</tr>
<tr>
<td>0091</td>
<td><code>&lt;ObjectType type=&quot;Enumeration&quot; value=&quot;Certificate&quot;/&gt;</code></td>
</tr>
<tr>
<td>0092</td>
<td><code>&lt;ObjectType type=&quot;Enumeration&quot; value=&quot;SymmetricKey&quot;/&gt;</code></td>
</tr>
<tr>
<td>0093</td>
<td><code>&lt;ObjectType type=&quot;Enumeration&quot; value=&quot;SecretData&quot;/&gt;</code></td>
</tr>
<tr>
<td>0094</td>
<td><code>&lt;ObjectType type=&quot;Enumeration&quot; value=&quot;PublicKey&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

4200780100000004000000010000000042007405000000040000000200000000000000010000000042000f010000003842005c050000000400000018000000004200790100000020420074050000000400000010000000200000000
7.3 Mandatory XML Profile Test Cases KMIP v1.2

7.3.1 MSGENC-XML-M-1-12 - 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.
<ResponseMessage>
  <ResponseHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1"/>
      <ProtocolVersionMinor type="Integer" value="2"/>
    </ProtocolVersion>
    <TimeStamp type="DateTime" value="2014-06-10T08:07:28+00:00"/>
    <BatchCount type="Integer" value="1"/>
  </ResponseHeader>
  <BatchItem>
    <Operation type="Enumeration" value="Query"/>
    <ResultStatus type="Enumeration" value="OperationFailed"/>
    <ResultReason type="Enumeration" value="ResponseTooLarge"/>
    <ResultMessage type="TextString" value="TOO_LARGE"/>
  </BatchItem>
</ResponseMessage>

42007b01000000a042007a0100000048420069010000002042006a0200000040000000100000000
42006b02000000040000000200000000042009290d00000008000000005396bc042000d020000004
00000001000000042000d010000000420005c05000000400000018000000042007f050000004
00000001000000042000d07e0500000040000000200000000420007d070000009544f5f4c415247
45000000000000

# TIME 1

<RequestMessage>
  <RequestHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1"/>
      <ProtocolVersionMinor type="Integer" value="2"/>
    </ProtocolVersion>
    <MaximumResponseSize type="Integer" value="2048"/>
    <BatchCount type="Integer" value="1"/>
  </RequestHeader>
  <BatchItem>
    <Operation type="Enumeration" value="Query"/>
    <RequestPayload>
      <QueryFunction type="Enumeration" value="QueryOperations"/>
      <QueryFunction type="Enumeration" value="QueryObjects"/>
    </RequestPayload>
  </BatchItem>
</RequestMessage>

42007801000000904200770100000048420069010000002042006a0200000040000000100000000
42006b02000000040000000200000000042009290d00000008000000005396bc042000d020000004
00000001000000042000d010000000420005c05000000400000018000000042007f050000004
00000001000000042000d07e0500000040000000200000000420007d070000009544f5f4c415247
45000000000000

<ResponseMessage>
  <ResponseHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1"/>
      <ProtocolVersionMinor type="Integer" value="2"/>
    </ProtocolVersion>
    <TimeStamp type="DateTime" value="2014-06-10T08:07:28+00:00"/>
    <BatchCount type="Integer" value="1"/>
  </ResponseHeader>
  <BatchItem>
    <Operation type="Enumeration" value="Query"/>
    <ResultStatus type="Enumeration" value="Success"/>
  </BatchItem>
</ResponseMessage>
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0062</td>
<td><code>&lt;ResponsePayload&gt;</code></td>
</tr>
<tr>
<td>0063</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Query&quot;/&gt;</code></td>
</tr>
<tr>
<td>0064</td>
<td><code>&lt;Operation type=&quot; Enumeration&quot; value=&quot;Locate&quot;/&gt;</code></td>
</tr>
<tr>
<td>0065</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Destroy&quot;/&gt;</code></td>
</tr>
<tr>
<td>0066</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Get&quot;/&gt;</code></td>
</tr>
<tr>
<td>0067</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Create&quot;/&gt;</code></td>
</tr>
<tr>
<td>0068</td>
<td><code>&lt;Operation type=&quot; Enumeration&quot; value=&quot;Register&quot;/&gt;</code></td>
</tr>
<tr>
<td>0069</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;GetAttributes&quot;/&gt;</code></td>
</tr>
<tr>
<td>0070</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;GetAttributeList&quot;/&gt;</code></td>
</tr>
<tr>
<td>0071</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;AddAttribute&quot;/&gt;</code></td>
</tr>
<tr>
<td>0072</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;ModifyAttribute&quot;/&gt;</code></td>
</tr>
<tr>
<td>0073</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;DeleteAttribute&quot;/&gt;</code></td>
</tr>
<tr>
<td>0074</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Activate&quot;/&gt;</code></td>
</tr>
<tr>
<td>0075</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Revoke&quot;/&gt;</code></td>
</tr>
<tr>
<td>0076</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Poll&quot;/&gt;</code></td>
</tr>
<tr>
<td>0077</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Cancel&quot;/&gt;</code></td>
</tr>
<tr>
<td>0078</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Check&quot;/&gt;</code></td>
</tr>
<tr>
<td>0079</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;GetUsageAllocation&quot;/&gt;</code></td>
</tr>
<tr>
<td>0080</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;CreateKeyPair&quot;/&gt;</code></td>
</tr>
<tr>
<td>0081</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;ReKey&quot;/&gt;</code></td>
</tr>
<tr>
<td>0082</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Archive&quot;/&gt;</code></td>
</tr>
<tr>
<td>0083</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Recover&quot;/&gt;</code></td>
</tr>
<tr>
<td>0084</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;ObtainLease&quot;/&gt;</code></td>
</tr>
<tr>
<td>0085</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;ReKeyKeyPair&quot;/&gt;</code></td>
</tr>
<tr>
<td>0086</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Certify&quot;/&gt;</code></td>
</tr>
<tr>
<td>0087</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;ReCertify&quot;/&gt;</code></td>
</tr>
<tr>
<td>0088</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;DiscoverVersions&quot;/&gt;</code></td>
</tr>
<tr>
<td>0089</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Notify&quot;/&gt;</code></td>
</tr>
<tr>
<td>0090</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Put&quot;/&gt;</code></td>
</tr>
<tr>
<td>0091</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;RNGRetrieve&quot;/&gt;</code></td>
</tr>
<tr>
<td>0092</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;RNGSeed&quot;/&gt;</code></td>
</tr>
<tr>
<td>0093</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Encrypt&quot;/&gt;</code></td>
</tr>
<tr>
<td>0094</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Decrypt&quot;/&gt;</code></td>
</tr>
<tr>
<td>0095</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Sign&quot;/&gt;</code></td>
</tr>
<tr>
<td>0096</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;SignatureVerify&quot;/&gt;</code></td>
</tr>
<tr>
<td>0097</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;MAC&quot;/&gt;</code></td>
</tr>
<tr>
<td>0098</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;MACVerify&quot;/&gt;</code></td>
</tr>
<tr>
<td>0099</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;Hash&quot;/&gt;</code></td>
</tr>
<tr>
<td>0100</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;CreateSplitKey&quot;/&gt;</code></td>
</tr>
<tr>
<td>0101</td>
<td><code>&lt;Operation type=&quot;Enumeration&quot; value=&quot;JoinSplitKey&quot;/&gt;</code></td>
</tr>
<tr>
<td>0102</td>
<td><code>&lt;ObjectType type=&quot;Enumeration&quot; value=&quot;Certificate&quot;/&gt;</code></td>
</tr>
<tr>
<td>0103</td>
<td><code>&lt;ObjectType type=&quot;Enumeration&quot; value=&quot;SymmetricKey&quot;/&gt;</code></td>
</tr>
<tr>
<td>0104</td>
<td><code>&lt;ObjectType type=&quot;Enumeration&quot; value=&quot;SecretData&quot;/&gt;</code></td>
</tr>
<tr>
<td>0105</td>
<td><code>&lt;ObjectType type=&quot;Enumeration&quot; value=&quot;PublicKey&quot;/&gt;</code></td>
</tr>
<tr>
<td>0106</td>
<td><code>&lt;ObjectType type=&quot;Enumeration&quot; value=&quot;PrivateKey&quot;/&gt;</code></td>
</tr>
<tr>
<td>0107</td>
<td><code>&lt;ObjectType type=&quot;Enumeration&quot; value=&quot;Template&quot;/&gt;</code></td>
</tr>
<tr>
<td>0108</td>
<td><code>&lt;ObjectType type=&quot;Enumeration&quot; value=&quot;OpaqueObject&quot;/&gt;</code></td>
</tr>
<tr>
<td>0109</td>
<td><code>&lt;ObjectType type=&quot;Enumeration&quot; value=&quot;SplitKey&quot;/&gt;</code></td>
</tr>
<tr>
<td>0110</td>
<td><code>&lt;ObjectType type=&quot;Enumeration&quot; value=&quot;PGPKey&quot;/&gt;</code></td>
</tr>
<tr>
<td>0111</td>
<td><code>&lt;/ResponsePayload&gt;</code></td>
</tr>
<tr>
<td>0112</td>
<td><code>&lt;/BatchItem&gt;</code></td>
</tr>
<tr>
<td>0113</td>
<td><code>&lt;/ResponseMessage&gt;</code></td>
</tr>
</tbody>
</table>
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.
4. SHALL support the HTTP Encoding conditions as specified in Section 2.5 of this profile.
5. Mandatory HTTPS Profile Test Cases KMIP v1.1 (3.2)

8.1.6 HTTPS Server KMIP v1.2 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 v1.2 (3.3)

8.2 JSON Profile

8.2.1 JSON Client KMIP v1.0 Profile Conformance

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

8.2.2 JSON Client KMIP v1.1 Profile Conformance

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

8.2.3 JSON Client KMIP v1.2 Profile Conformance

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

8.2.4 JSON Server KMIP v1.0 Profile Conformance

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

8.2.5 JSON Server KMIP v1.1 Profile Conformance

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

8.2.6 JSON Server KMIP v1.2 Profile Conformance

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

8.3 XML Profile

8.3.1 XML Client KMIP v1.0 Profile Conformance

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

8.3.2 XML Client KMIP v1.1 Profile Conformance

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

8.3.3 XML Client KMIP v1.2 Profile Conformance

KMIP client implementations conformant to this profile:

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

8.3.4 XML Server KMIP v1.0 Profile Conformance

KMIP server implementations conformant to this profile:

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

8.3.5 XML Server KMIP v1.1 Profile Conformance

KMIP server implementations conformant to this profile:

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

8.3.6 XML Server KMIP v1.2 Profile Conformance

KMIP server implementations conformant to this profile:

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

8.4 Permitted Test Case Variations

Whilst the test cases provided in this Profile define the allowed request and response content, some 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 Bagner, 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
Jin Wong, QuintessenceLabs
Sameer Yami, Thales e-Security
Peter Yee, EMC Corporation
Krishna Yellepeddy, IBM
Catherine Ying, SafeNet, Inc.
Tatu Ylonen, SSH Communications Security (Tectia Corp)
Michael Yoder, Vormetric, Inc.
Magda Zdunkiewicz, Cryptsoft
Peter Zelechoski, Election Systems & Software
# Appendix B. KMIP Specification Cross Reference

<table>
<thead>
<tr>
<th>Reference Term</th>
<th>KMIP 1.0</th>
<th>KMIP 1.1</th>
<th>KMIP 1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1 Introduction</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-Normative References</td>
<td>1.3.</td>
<td>1.3.</td>
<td>1.3.</td>
</tr>
<tr>
<td>Normative References</td>
<td>1.2.</td>
<td>1.2.</td>
<td>1.2.</td>
</tr>
<tr>
<td>Terminology</td>
<td>1.1.</td>
<td>1.1.</td>
<td>1.1.</td>
</tr>
<tr>
<td><strong>2 Objects</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>2.1.1.</td>
<td>2.1.1.</td>
<td>2.1.1.</td>
</tr>
<tr>
<td>Base Objects</td>
<td>2.1.</td>
<td>2.1.</td>
<td>2.1.</td>
</tr>
<tr>
<td>Certificate</td>
<td>2.2.1.</td>
<td>2.2.1.</td>
<td>2.2.1.</td>
</tr>
<tr>
<td>Credential</td>
<td>2.1.2.</td>
<td>2.1.2.</td>
<td>2.1.2.</td>
</tr>
<tr>
<td>Data</td>
<td>-</td>
<td>-</td>
<td>2.1.10.</td>
</tr>
<tr>
<td>Data Length</td>
<td>-</td>
<td>-</td>
<td>2.1.11.</td>
</tr>
<tr>
<td>Extension Information</td>
<td>-</td>
<td>2.1.9.</td>
<td>2.1.9.</td>
</tr>
<tr>
<td>Key Block</td>
<td>2.1.3.</td>
<td>2.1.3.</td>
<td>2.1.3.</td>
</tr>
<tr>
<td>Key Value</td>
<td>2.1.4.</td>
<td>2.1.4.</td>
<td>2.1.4.</td>
</tr>
<tr>
<td>Key Wrapping Data</td>
<td>2.1.5.</td>
<td>2.1.5.</td>
<td>2.1.5.</td>
</tr>
<tr>
<td>Key Wrapping Specification</td>
<td>2.1.6.</td>
<td>2.1.6.</td>
<td>2.1.6.</td>
</tr>
<tr>
<td>MAC Data</td>
<td>-</td>
<td>-</td>
<td>2.1.13.</td>
</tr>
<tr>
<td>Managed Objects</td>
<td>2.2.</td>
<td>2.2.</td>
<td>2.2.</td>
</tr>
<tr>
<td>Opaque Object</td>
<td>2.2.8.</td>
<td>2.2.8.</td>
<td>2.2.8.</td>
</tr>
<tr>
<td>PGP Key</td>
<td>-</td>
<td>-</td>
<td>2.2.9.</td>
</tr>
<tr>
<td>Private Key</td>
<td>2.2.4.</td>
<td>2.2.4.</td>
<td>2.2.4.</td>
</tr>
<tr>
<td>Public Key</td>
<td>2.2.3.</td>
<td>2.2.3.</td>
<td>2.2.3.</td>
</tr>
<tr>
<td>Secret Data</td>
<td>2.2.7.</td>
<td>2.2.7.</td>
<td>2.2.7.</td>
</tr>
<tr>
<td>Signature Data</td>
<td>-</td>
<td>-</td>
<td>2.1.12.</td>
</tr>
<tr>
<td>Split Key</td>
<td>2.2.5.</td>
<td>2.2.5.</td>
<td>2.2.5.</td>
</tr>
<tr>
<td>Symmetric Key</td>
<td>2.2.2.</td>
<td>2.2.2.</td>
<td>2.2.2.</td>
</tr>
<tr>
<td>Template</td>
<td>2.2.6.</td>
<td>2.2.6.</td>
<td>2.2.6.</td>
</tr>
<tr>
<td>Template-Attribute Structures</td>
<td>2.1.8.</td>
<td>2.1.8.</td>
<td>2.1.8.</td>
</tr>
<tr>
<td>Transparent DH Private Key</td>
<td>2.1.7.6.</td>
<td>2.1.7.6.</td>
<td>2.1.7.6.</td>
</tr>
<tr>
<td>Transparent DH Public Key</td>
<td>2.1.7.7.</td>
<td>2.1.7.7.</td>
<td>2.1.7.7.</td>
</tr>
<tr>
<td>Transparent DSA Private Key</td>
<td>2.1.7.2.</td>
<td>2.1.7.2.</td>
<td>2.1.7.2.</td>
</tr>
<tr>
<td>Transparent DSA Public Key</td>
<td>2.1.7.3.</td>
<td>2.1.7.3.</td>
<td>2.1.7.3.</td>
</tr>
<tr>
<td>Transparent ECDH Private Key</td>
<td>2.1.7.10.</td>
<td>2.1.7.10.</td>
<td>2.1.7.10.</td>
</tr>
<tr>
<td>Transparent ECDH Public Key</td>
<td>2.1.7.11.</td>
<td>2.1.7.11.</td>
<td>2.1.7.11.</td>
</tr>
<tr>
<td>Transparent ECDSA Private Key</td>
<td>2.1.7.8.</td>
<td>2.1.7.8.</td>
<td>2.1.7.8.</td>
</tr>
<tr>
<td>Transparent ECDSA Public Key</td>
<td>2.1.7.9.</td>
<td>2.1.7.9.</td>
<td>2.1.7.9.</td>
</tr>
<tr>
<td>Transparent ECMQV Private Key</td>
<td>2.1.7.12.</td>
<td>2.1.7.12.</td>
<td>2.1.7.12.</td>
</tr>
<tr>
<td>Transparent ECMQV Public Key</td>
<td>2.1.7.13.</td>
<td>2.1.7.13.</td>
<td>2.1.7.13.</td>
</tr>
<tr>
<td>Transparent Key Structures</td>
<td>2.1.7.</td>
<td>2.1.7.</td>
<td>2.1.7.</td>
</tr>
<tr>
<td>Transparent RSA Private Key</td>
<td>2.1.7.4.</td>
<td>2.1.7.4.</td>
<td>2.1.7.4.</td>
</tr>
<tr>
<td>Transparent RSA Public Key</td>
<td>2.1.7.5.</td>
<td>2.1.7.5.</td>
<td>2.1.7.5.</td>
</tr>
<tr>
<td>Transparent Symmetric Key</td>
<td>2.1.7.1.</td>
<td>2.1.7.1.</td>
<td>2.1.7.1.</td>
</tr>
<tr>
<td><strong>3 Attributes</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activation Date</td>
<td>3.19.</td>
<td>3.24.</td>
<td>3.24.</td>
</tr>
<tr>
<td>Alternative Name</td>
<td>-</td>
<td>-</td>
<td>3.40.</td>
</tr>
<tr>
<td>Application Specific Information</td>
<td>3.30.</td>
<td>3.36.</td>
<td>3.36.</td>
</tr>
<tr>
<td>Archive Date</td>
<td>3.27.</td>
<td>3.32.</td>
<td>3.32.</td>
</tr>
<tr>
<td>Reference Term</td>
<td>KMIP 1.0</td>
<td>KMIP 1.1</td>
<td>KMIP 1.2</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>Attributes</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Certificate Identifier</td>
<td>3.9.</td>
<td>3.13.</td>
<td>3.13.</td>
</tr>
<tr>
<td>Certificate Issuer</td>
<td>3.11.</td>
<td>3.15.</td>
<td>3.15.</td>
</tr>
<tr>
<td>Certificate Length</td>
<td>-</td>
<td>3.9.</td>
<td>3.9.</td>
</tr>
<tr>
<td>Certificate Subject</td>
<td>3.10.</td>
<td>3.14.</td>
<td>3.14.</td>
</tr>
<tr>
<td>Certificate Type</td>
<td>3.8.</td>
<td>3.8.</td>
<td>3.8.</td>
</tr>
<tr>
<td>Compromise Date</td>
<td>3.25.</td>
<td>3.30.</td>
<td>3.30.</td>
</tr>
<tr>
<td>Compromise Occurrence Date</td>
<td>3.24.</td>
<td>3.29.</td>
<td>3.29.</td>
</tr>
<tr>
<td>Contact Information</td>
<td>3.31.</td>
<td>3.37.</td>
<td>3.37.</td>
</tr>
<tr>
<td>Cryptographic Algorithm</td>
<td>3.4.</td>
<td>3.4.</td>
<td>3.4.</td>
</tr>
<tr>
<td>Cryptographic Domain Parameters</td>
<td>3.7.</td>
<td>3.7.</td>
<td>3.7.</td>
</tr>
<tr>
<td>Cryptographic Length</td>
<td>3.5.</td>
<td>3.5.</td>
<td>3.5.</td>
</tr>
<tr>
<td>Cryptographic Parameters</td>
<td>3.6.</td>
<td>3.6.</td>
<td>3.6.</td>
</tr>
<tr>
<td>Custom Attribute</td>
<td>3.33.</td>
<td>3.39.</td>
<td>3.39.</td>
</tr>
<tr>
<td>Deactivation Date</td>
<td>3.22.</td>
<td>3.27.</td>
<td>3.27.</td>
</tr>
<tr>
<td>Default Operation Policy</td>
<td>3.13.2.</td>
<td>3.18.2.</td>
<td>3.18.2.</td>
</tr>
<tr>
<td>Default Operation Policy for Certificates and Public Key Objects</td>
<td>3.13.2.2.</td>
<td>3.18.2.2.</td>
<td>3.18.2.2.</td>
</tr>
<tr>
<td>Default Operation Policy for Secret Objects</td>
<td>3.13.2.1.</td>
<td>3.18.2.1.</td>
<td>3.18.2.1.</td>
</tr>
<tr>
<td>Default Operation Policy for Template Objects</td>
<td>3.13.2.3.</td>
<td>3.18.2.3.</td>
<td>3.18.2.3.</td>
</tr>
<tr>
<td>Destroy Date</td>
<td>3.23.</td>
<td>3.28.</td>
<td>3.28.</td>
</tr>
<tr>
<td>Digest</td>
<td>3.12.</td>
<td>3.17.</td>
<td>3.17.</td>
</tr>
<tr>
<td>Digital Signature Algorithm</td>
<td>-</td>
<td>3.16.</td>
<td>3.16.</td>
</tr>
<tr>
<td>Fresh</td>
<td>-</td>
<td>3.34.</td>
<td>3.34.</td>
</tr>
<tr>
<td>Initial Date</td>
<td>3.18.</td>
<td>3.23.</td>
<td>3.23.</td>
</tr>
<tr>
<td>Key Value Location</td>
<td>-</td>
<td>-</td>
<td>3.41.</td>
</tr>
<tr>
<td>Key Value Present</td>
<td>-</td>
<td>-</td>
<td>3.41.</td>
</tr>
<tr>
<td>Last Change Date</td>
<td>3.32.</td>
<td>3.38.</td>
<td>3.38.</td>
</tr>
<tr>
<td>Lease Time</td>
<td>3.15.</td>
<td>3.20.</td>
<td>3.20.</td>
</tr>
<tr>
<td>Link</td>
<td>3.29.</td>
<td>3.35.</td>
<td>3.35.</td>
</tr>
<tr>
<td>Name</td>
<td>3.2.</td>
<td>3.2.</td>
<td>3.2.</td>
</tr>
<tr>
<td>Object Group</td>
<td>3.28.</td>
<td>3.33.</td>
<td>3.33.</td>
</tr>
<tr>
<td>Object Type</td>
<td>3.3.</td>
<td>3.3.</td>
<td>3.3.</td>
</tr>
<tr>
<td>Operation Policy Name</td>
<td>3.13.</td>
<td>3.18.</td>
<td>3.18.</td>
</tr>
<tr>
<td>Operations outside of operation policy control</td>
<td>3.13.1.</td>
<td>3.18.1.</td>
<td>3.18.1.</td>
</tr>
<tr>
<td>Original Creation Date</td>
<td>-</td>
<td>-</td>
<td>3.43.</td>
</tr>
<tr>
<td>Process Start Date</td>
<td>3.20.</td>
<td>3.25.</td>
<td>3.25.</td>
</tr>
<tr>
<td>Protect Stop Date</td>
<td>3.21.</td>
<td>3.26.</td>
<td>3.26.</td>
</tr>
<tr>
<td>Revocation Reason</td>
<td>3.26.</td>
<td>3.31.</td>
<td>3.31.</td>
</tr>
<tr>
<td>State</td>
<td>3.17.</td>
<td>3.22.</td>
<td>3.22.</td>
</tr>
<tr>
<td>Unique Identifier</td>
<td>3.1.</td>
<td>3.1.</td>
<td>3.1.</td>
</tr>
<tr>
<td>Usage Limits</td>
<td>3.16.</td>
<td>3.21.</td>
<td>3.21.</td>
</tr>
<tr>
<td>X.509 Certificate Identifier</td>
<td>-</td>
<td>3.10.</td>
<td>3.10.</td>
</tr>
<tr>
<td>X.509 Certificate Subject</td>
<td>-</td>
<td>3.11.</td>
<td>3.11.</td>
</tr>
</tbody>
</table>

4 Client-to-Server Operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>KMIP 1.0</th>
<th>KMIP 1.1</th>
<th>KMIP 1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate</td>
<td>4.18.</td>
<td>4.19.</td>
<td>4.19.</td>
</tr>
<tr>
<td>Archive</td>
<td>4.21.</td>
<td>4.22.</td>
<td>4.22.</td>
</tr>
<tr>
<td>Cancel</td>
<td>4.25.</td>
<td>4.27.</td>
<td>4.27.</td>
</tr>
<tr>
<td>Certify</td>
<td>4.6.</td>
<td>4.7.</td>
<td>4.7.</td>
</tr>
<tr>
<td>Check</td>
<td>4.9.</td>
<td>4.10.</td>
<td>4.10.</td>
</tr>
<tr>
<td>Create</td>
<td>4.1.</td>
<td>4.1.</td>
<td>4.1.</td>
</tr>
<tr>
<td>Create Key Pair</td>
<td>4.2.</td>
<td>4.2.</td>
<td>4.2.</td>
</tr>
<tr>
<td>Reference Term</td>
<td>KMIP 1.0</td>
<td>KMIP 1.1</td>
<td>KMIP 1.2</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>Create Split Key</td>
<td>-</td>
<td>-</td>
<td>4.38.</td>
</tr>
<tr>
<td>Decrypt</td>
<td>-</td>
<td>-</td>
<td>4.30.</td>
</tr>
<tr>
<td>Delete Attribute</td>
<td>4.15.</td>
<td>4.16.</td>
<td>4.16.</td>
</tr>
<tr>
<td>Derive Key</td>
<td>4.5.</td>
<td>4.6.</td>
<td>4.6.</td>
</tr>
<tr>
<td>Destroy</td>
<td>4.20.</td>
<td>4.21.</td>
<td>4.21.</td>
</tr>
<tr>
<td>Encrypt</td>
<td>-</td>
<td>-</td>
<td>4.29.</td>
</tr>
<tr>
<td>Get</td>
<td>4.10.</td>
<td>4.11.</td>
<td>4.11.</td>
</tr>
<tr>
<td>Get Attributes</td>
<td>4.11.</td>
<td>4.12.</td>
<td>4.12.</td>
</tr>
<tr>
<td>Get Usage Allocation</td>
<td>4.17.</td>
<td>4.18.</td>
<td>4.18.</td>
</tr>
<tr>
<td>Hash</td>
<td>-</td>
<td>-</td>
<td>4.37.</td>
</tr>
<tr>
<td>Join Split Key</td>
<td>-</td>
<td>-</td>
<td>4.39.</td>
</tr>
<tr>
<td>Locate</td>
<td>4.8.</td>
<td>4.9.</td>
<td>4.9.</td>
</tr>
<tr>
<td>MAC</td>
<td>-</td>
<td>-</td>
<td>4.33.</td>
</tr>
<tr>
<td>MAC Verify</td>
<td>-</td>
<td>-</td>
<td>4.34.</td>
</tr>
<tr>
<td>Modify Attribute</td>
<td>4.14.</td>
<td>4.15.</td>
<td>4.15.</td>
</tr>
<tr>
<td>Obtain Lease</td>
<td>4.16.</td>
<td>4.17.</td>
<td>4.17.</td>
</tr>
<tr>
<td>Poll</td>
<td>4.26.</td>
<td>4.28.</td>
<td>4.28.</td>
</tr>
<tr>
<td>Query</td>
<td>4.24.</td>
<td>4.25.</td>
<td>4.25.</td>
</tr>
<tr>
<td>Re-certify</td>
<td>4.7.</td>
<td>4.8.</td>
<td>4.8.</td>
</tr>
<tr>
<td>Recover</td>
<td>4.22.</td>
<td>4.23.</td>
<td>4.23.</td>
</tr>
<tr>
<td>Register</td>
<td>4.3.</td>
<td>4.3.</td>
<td>4.3.</td>
</tr>
<tr>
<td>Re-key</td>
<td>4.4.</td>
<td>4.4.</td>
<td>4.4.</td>
</tr>
<tr>
<td>Re-key Key Pair</td>
<td>-</td>
<td>4.5.</td>
<td>4.5.</td>
</tr>
<tr>
<td>Revoke</td>
<td>4.19.</td>
<td>4.20.</td>
<td>4.20.</td>
</tr>
<tr>
<td>RNG Retrieve</td>
<td>-</td>
<td>-</td>
<td>4.35.</td>
</tr>
<tr>
<td>RNG Seed</td>
<td>-</td>
<td>-</td>
<td>4.36.</td>
</tr>
<tr>
<td>Sign</td>
<td>-</td>
<td>-</td>
<td>4.31.</td>
</tr>
<tr>
<td>Signature Verify</td>
<td>-</td>
<td>-</td>
<td>4.32.</td>
</tr>
<tr>
<td>Validate</td>
<td>4.23.</td>
<td>4.24.</td>
<td>4.24.</td>
</tr>
</tbody>
</table>

## 5 Server-to-Client Operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>KMIP 1.0</th>
<th>KMIP 1.1</th>
<th>KMIP 1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notify</td>
<td>5.1.</td>
<td>5.1.</td>
<td>5.1.</td>
</tr>
<tr>
<td>Put</td>
<td>5.2.</td>
<td>5.2.</td>
<td>5.2.</td>
</tr>
</tbody>
</table>

## 6 Message Contents

<table>
<thead>
<tr>
<th>Message Content</th>
<th>KMIP 1.0</th>
<th>KMIP 1.1</th>
<th>KMIP 1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asynchronous Correlation Value</td>
<td>6.8.</td>
<td>6.8.</td>
<td>6.8.</td>
</tr>
<tr>
<td>Asynchronous Indicator</td>
<td>6.7.</td>
<td>6.7.</td>
<td>6.7.</td>
</tr>
<tr>
<td>Attestation Capable Indicator</td>
<td>-</td>
<td>-</td>
<td>6.17.</td>
</tr>
<tr>
<td>Batch Item</td>
<td>6.15.</td>
<td>6.15.</td>
<td>6.15.</td>
</tr>
<tr>
<td>Maximum Response Size</td>
<td>6.3.</td>
<td>6.3.</td>
<td>6.3.</td>
</tr>
<tr>
<td>Message Extension</td>
<td>6.16.</td>
<td>6.16.</td>
<td>6.16.</td>
</tr>
<tr>
<td>Operation</td>
<td>6.2.</td>
<td>6.2.</td>
<td>6.2.</td>
</tr>
<tr>
<td>Result Message</td>
<td>6.11.</td>
<td>6.11.</td>
<td>6.11.</td>
</tr>
<tr>
<td>Result Reason</td>
<td>6.10.</td>
<td>6.10.</td>
<td>6.10.</td>
</tr>
<tr>
<td>Result Status</td>
<td>6.9.</td>
<td>6.9.</td>
<td>6.9.</td>
</tr>
<tr>
<td>Time Stamp</td>
<td>6.5.</td>
<td>6.5.</td>
<td>6.5.</td>
</tr>
<tr>
<td>Unique Batch Item ID</td>
<td>6.4.</td>
<td>6.4.</td>
<td>6.4.</td>
</tr>
</tbody>
</table>

## 7 Message Format
### Reference Term

<table>
<thead>
<tr>
<th>Reference Term</th>
<th>KMIP 1.0</th>
<th>KMIP 1.1</th>
<th>KMIP 1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Structure</td>
<td>7.1</td>
<td>7.1</td>
<td>7.1</td>
</tr>
<tr>
<td>Operations</td>
<td>7.2</td>
<td>7.2</td>
<td>7.2</td>
</tr>
</tbody>
</table>

### 8 Authentication

- Authentication

### 9 Message Encoding

- Alternative Name Type Enumeration
- Attestation Type Enumeration
- Batch Error Continuation Option Enumeration
  - 9.1.3.2.29.
  - 9.1.3.2.30.
  - 9.1.3.2.30.
- Bit Masks
  - 9.1.3.3.
  - 9.1.3.3.
  - 9.1.3.3.
- Block Cipher Mode Enumeration
  - 9.1.3.2.13.
  - 9.1.3.2.14.
  - 9.1.3.2.14.
- Cancellation Result Enumeration
  - 9.1.3.2.24.
  - 9.1.3.2.25.
  - 9.1.3.2.25.
- Certificate Request Type Enumeration
  - 9.1.3.2.21.
  - 9.1.3.2.22.
  - 9.1.3.2.22.
- Certificate Type Enumeration
  - 9.1.3.2.6.
  - 9.1.3.2.6.
  - 9.1.3.2.6.
- Credential Type Enumeration
  - 9.1.3.2.1.
  - 9.1.3.2.1.
  - 9.1.3.2.1.
- Cryptographic Algorithm Enumeration
  - 9.1.3.2.12.
  - 9.1.3.2.13.
  - 9.1.3.2.13.
- Cryptographic Usage Mask
  - 9.1.3.3.1.
  - 9.1.3.3.1.
  - 9.1.3.3.1.
- Defined Values
  - 9.1.3.
  - 9.1.3.
  - 9.1.3.
- Derivation Method Enumeration
  - 9.1.3.2.20.
  - 9.1.3.2.21.
  - 9.1.3.2.21.
- Digital Signature Algorithm Enumeration
  - 9.1.3.2.7.
  - 9.1.3.2.7.
- Encoding Option Enumeration
  - 9.1.3.2.32.
  - 9.1.3.2.32.
- Enumerations
  - 9.1.3.2.
  - 9.1.3.2.
  - 9.1.3.2.
- Examples
  - 9.1.2.
  - 9.1.2.
  - 9.1.2.
- Hashing Algorithm Enumeration
  - 9.1.3.2.15.
  - 9.1.3.2.16.
  - 9.1.3.2.16.
- Item Length
  - 9.1.1.3.
  - 9.1.1.3.
  - 9.1.1.3.
- Item Tag
  - 9.1.1.1.
  - 9.1.1.1.
  - 9.1.1.1.
- Item Type
  - 9.1.1.2.
  - 9.1.1.2.
  - 9.1.1.2.
- Item Value
  - 9.1.1.4.
  - 9.1.1.4.
  - 9.1.1.4.
- Key Compression Type Enumeration
  - 9.1.3.2.2.
  - 9.1.3.2.2.
  - 9.1.3.2.2.
- Key Format Type Enumeration
  - 9.1.3.2.3.
  - 9.1.3.2.3.
  - 9.1.3.2.3.
- Key Role Type Enumeration
  - 9.1.3.2.16.
  - 9.1.3.2.17.
  - 9.1.3.2.17.
- Key Value Location Type Enumeration
  - -
  - -
  - -
- Link Type Enumeration
  - 9.1.3.2.19.
  - 9.1.3.2.20.
  - 9.1.3.2.20.
- Name Type Enumeration
  - 9.1.3.2.10.
  - 9.1.3.2.11.
  - 9.1.3.2.11.
- Object Group Member Enumeration
  - -
  - 9.1.3.2.33.
  - 9.1.3.2.33.
- Object Type Enumeration
  - 9.1.3.2.11.
  - 9.1.3.2.12.
  - 9.1.3.2.12.
- Opaque Data Type Enumeration
  - 9.1.3.2.9.
  - 9.1.3.2.10.
  - 9.1.3.2.10.
- Operation Enumeration
  - 9.1.3.2.26.
  - 9.1.3.2.27.
  - 9.1.3.2.27.
- Padding Method Enumeration
  - 9.1.3.2.14.
  - 9.1.3.2.15.
  - 9.1.3.2.15.
- Put Function Enumeration
  - 9.1.3.2.25.
  - 9.1.3.2.26.
  - 9.1.3.2.26.
- Query Function Enumeration
  - 9.1.3.2.23.
  - 9.1.3.2.24.
  - 9.1.3.2.24.
- Recommended Curve Enumeration for ECDSA, ECDH, and ECMQV
  - 9.1.3.2.5.
  - 9.1.3.2.5.
  - 9.1.3.2.5.
- Result Reason Enumeration
  - 9.1.3.2.28.
  - 9.1.3.2.29.
  - 9.1.3.2.29.
- Result Status Enumeration
  - 9.1.3.2.27.
  - 9.1.3.2.28.
  - 9.1.3.2.28.
- Revocation Reason Code Enumeration
  - 9.1.3.2.18.
  - 9.1.3.2.19.
  - 9.1.3.2.19.
- Secret Data Type Enumeration
  - 9.1.3.2.8.
  - 9.1.3.2.9.
  - 9.1.3.2.9.
- Split Key Method Enumeration
  - 9.1.3.2.7.
  - 9.1.3.2.8.
  - 9.1.3.2.8.
- State Enumeration
  - 9.1.3.2.17.
  - 9.1.3.2.18.
  - 9.1.3.2.18.
- Storage Status Mask
  - 9.1.3.3.2.
  - 9.1.3.3.2.
  - 9.1.3.3.2.
- Tags
  - 9.1.3.1.
  - 9.1.3.1.
  - 9.1.3.1.
- TTLV Encoding
  - 9.1.
  - 9.1.
  - 9.1.
- TTLV Encoding Fields
  - 9.1.1.
  - 9.1.1.
  - 9.1.1.
- Usage Limits Unit Enumeration
  - 9.1.3.2.30.
  - 9.1.3.2.31.
  - 9.1.3.2.31.
<table>
<thead>
<tr>
<th>Reference Term</th>
<th>KMIP 1.0</th>
<th>KMIP 1.1</th>
<th>KMIP 1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validity Indicator Enumeration</td>
<td>9.1.3.2.22.</td>
<td>9.1.3.2.23.</td>
<td>9.1.3.2.23.</td>
</tr>
<tr>
<td>Wrapping Method Enumeration</td>
<td>9.1.3.2.4.</td>
<td>9.1.3.2.4.</td>
<td>9.1.3.2.4.</td>
</tr>
<tr>
<td>XML Encoding</td>
<td>9.2.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

10 Transport

| Transport | 10 | 10 | 10 |

12 KMIP Server and Client Implementation Conformance

| Conformance clauses for a KMIP Server | 12.1. | -   | -   |
| KMIP Client Implementation Conformance | -    | 12.2. | 12.2. |
| KMIP Server Implementation Conformance | -    | 12.1. | 12.1. |
# Appendix C. Revision History

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