
Project Specification Draft 02
14 November 2019

This stage:
https://docs.oasis-open-projects.org/oslc-op/qm/v2.1/psd02/quality-management-spec.html (Authoritative)
https://docs.oasis-open-projects.org/oslc-op/qm/v2.1/psd02/quality-management-spec.pdf

Previous stage:
N/A

Latest stage:
https://docs.oasis-open-projects.org/oslc-op/qm/v2.1/quality-management-spec.html (Authoritative)
https://docs.oasis-open-projects.org/oslc-op/qm/v2.1/quality-management-spec.pdf

Latest version:
https://open-services.net/spec/qm/latest

Latest editor's draft:
https://open-services.net/spec/qm/latest-draft

Open Project:
OASIS Open Services for Lifecycle Collaboration (OSLC) OP

Project Chairs:
Jim Amsden (jamsden@us.ibm.com), IBM
Andrii Berezovskyi (andriib@kth.se), KTH

Editors:
Jim Amsden (jamsden@us.ibm.com), IBM
Gray Bachelor (gray_bachelor@uk.ibm.com), IBM

Additional components:
This specification is one component of a Work Product that also includes:


RDF Namespaces:
http://open-services.net/ns/core/qm#
Abstract:
This specification defines the OSLC Quality Management domain, a RESTful web services interface for the management of product, service or software quality artefacts, activities, tasks and relationships between those and related resources such as requirements, defects, change requests or architectural resources. To support these scenarios, this specification defines a set of HTTP-based RESTful interfaces in terms of HTTP methods: GET, POST, PUT and DELETE, HTTP response codes, content type handling and resource formats.

Status:
This document was last revised or approved by the OASIS Open Services for Lifecycle Collaboration (OSLC) OP on the above date. The level of approval is also listed above. Check the “Latest stage” location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Open Project are listed at https://open-services.net/about/.

Comments on this work can be provided by opening issues in the project repository or by sending email to the project’s public comment list oslc-op@lists.oasis-open-projects.org.

Note that any machine-readable content (Computer Language Definitions) declared Normative for this Work Product is provided in separate plain text files. In the event of a discrepancy between any such plain text file and display content in the Work Product’s prose narrative document(s), the content in the separate plain text file prevails.

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

[OSLC-QM-2.1-Part1]
Latest stage: https://docs.oasis-open-projects.org/oslc-op/qm/v2.1/quality-management-spec.html.
Table of Contents

1. Introduction
   1.1 Overview
   1.2 Terminology
   1.3 References
      1.3.1 Normative references
      1.3.2 Informative references
   1.4 Typographical Conventions and Use of RFC Terms

2. Base Requirements
   2.1 Base Requirements
   2.2 Specification Versioning
   2.3 Namespaces
   2.4 Resource Formats
      2.4.1 Content Negotiation
   2.5 Authentication
   2.6 Error Responses
   2.7 Pagination
   2.8 Requesting and Updating Properties
      2.8.1 Requesting a Subset of Properties
      2.8.2 Updating a Subset of Properties
      2.8.3 Updating Multi-Valued Properties
   2.9 Labels for Relationships

3. Vocabulary Terms and Constraints

4. QM Server Capabilities
   4.1 Resource Shapes
   4.2 Service Provider Resources
   4.3 Creation Factories
   4.4 Query Capabilities
   4.5 Delegated UIs

5. Conformance

Appendix A. Version Compatibility with 2.0 Specifications
Appendix B. Acknowledgements
1. Introduction

1.1 Overview

This section is non-normative.

This specification builds on the OSLC Core Specification to define the resources and operations supported by an Open Services for Lifecycle Collaboration (OSLC) Quality Management (QM) provider.

Quality Management resources define the test plans, test cases, and test results of the software delivery lifecycle. They represent individual resources along with their relationships to other shared resource types such change requests and requirements. The intent of this specification is to define the set of HTTP-based RESTful interfaces in terms of HTTP methods: GET, POST, PUT and DELETE, HTTP response codes, mime type handling and resource formats. The capabilities of the interface definitions are driven by key integration scenarios and therefore don't represent a complete setup of operations on resources or resource types. The resource formats and operations may not match exactly the native models supported by quality management service providers but are intended to be compatible with them.

A key approach to supporting these scenarios is to delegate operations, as driven by service provider contributed user interfaces, as much as possible and not require a service provider to expose its complete data model and application logic.

1.2 Terminology

Terminology is based on OSLC Core Overview [OSLCCore3], W3C Linked Data Platform [LDP], W3C’s Architecture of the World Wide Web [WEBARCH], and Hyper-text Transfer Protocol [HTTP11].

Service Provider

An implementation of the OSLC Quality Management specifications as a server. OSLC QM clients consume these services.

Quality Management Resource

A resource managed by an OSLC QM service provider. The types of resources defined by this specification are Test Plan, Test Case, Test Script, Test Execution Record, and Test Result.

Client

An implementation of the OSLC Quality Management specifications as a client. OSLC QM Clients consume services provided by servers.

Server

An implementation of the OSLC Quality Management specifications as a server. OSLC QM clients consume services provided by Servers. The use of the terms Client and Server are intended to distinguish typical consumers and providers of OSLC resources in a distributed environment based on REST. A particular application component could be a client for some OSLC domain services and a server for the same or another domain.

Test Plan Resource

Defines the overall process and strategy for testing a system.

Test Case Resource

Defines the criteria which determine whether a system exhibits the correct behavior under a specific set of circumstances.

Test Script Resource
Defines a program or list of steps used to conduct a test

**Test Execution Record Resource**
Planning for execution of a test

**Test Result Resource**
Describes the outcome of attempting to execute a test

### 1.3 References

#### 1.3.1 Normative references

[OSLCCore2]
Dave Johnson; S. Speicher. *OSLC Core Specification 2.0*. Finalized. URL: [https://archive.open-services.net/bin/view/Main/OslcCoreSpecification](https://archive.open-services.net/bin/view/Main/OslcCoreSpecification)

[OSLCCore3]

[OSLCCoreVocab]

[OSLCPreview]

[OSLCQM1]
Paul McMahan. *OSLC Quality Management 1.0 Specification*. Finalized. URL: [https://archive.open-services.net/bin/view/Main/QmSpecificationV1.html](https://archive.open-services.net/bin/view/Main/QmSpecificationV1.html)

[OSLCShapes]

[OpenIDConnect]
*OpenID Connect*. URL: [http://openid.net/connect/](http://openid.net/connect/)

[RFC2119]

#### 1.3.2 Informative references

[HTTP11]
1.4 Typographical Conventions and Use of RFC Terms

As well as sections marked as non-normative, all authoring guidelines, diagrams, examples, and notes in this specification are non-normative. Everything else in this specification is normative.

The key words MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this specification are to be interpreted as described in [RFC2119].

Conventions
2. Base Requirements

The following sub-sections define the mandatory and optional requirements for an OSLC Quality Management (OSLC QM) server.

2.1 Base Requirements

This specification is based on [OSLCCore3]. OSLC QM servers **MUST** be compliant with both the core specification, **MUST** follow all the mandatory requirements in the normative sections of this specification, and **SHOULD** follow all the guidelines and recommendations in both these specifications. [qm-1]

An OSLC QM server **MUST** implement the domain vocabulary defined in OSLC Quality Management Version 2.1. Part 2: Vocabulary [qm-2]

The following table summarizes the requirements from OSLC Core Specification as well as some additional requirements specific to the QM domain. Note that this specification further restricts some of the requirements for OSLC Core Specification. See subsequent sections in this specification or the OSLC Core Specification to get further details on each of these requirements.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown properties and content</td>
<td>OSLC services <strong>MAY</strong> ignore unknown content and OSLC clients <strong>MUST</strong> preserve unknown content [qm-3]</td>
</tr>
<tr>
<td>Resource Operations</td>
<td>OSLC service <strong>MUST</strong> support resource operations via standard HTTP operations [qm-4]</td>
</tr>
<tr>
<td>Resource Paging</td>
<td>OSLC services <strong>MAY</strong> provide paging for resources but only when specifically requested by client [qm-5]</td>
</tr>
<tr>
<td>Partial Resource Representations</td>
<td>OSLC services <strong>MUST</strong> support request for a subset of a resource’s properties via the oslc.properties URL parameter retrieval via HTTP GET [qm-6]</td>
</tr>
<tr>
<td>Partial Update</td>
<td>OSLC services <strong>MAY</strong> support partial update of resources using patch semantics and <strong>MAY</strong> support via HTTP PUT [qm-7]</td>
</tr>
<tr>
<td>Discovery</td>
<td>OSLC servers <strong>MUST</strong> support OSLC Core 3.0 Discovery, <strong>MAY</strong> provide a ServiceProviderCatalog and <strong>SHOULD</strong> provide a ServiceProvider resource for Core v2 compatibility. [qm-8]</td>
</tr>
<tr>
<td>Creation Factories</td>
<td>OSLC servers <strong>MUST</strong> provide LDPC creation factories to enable resource creation of Quality Management resources via HTTP POST [qm-9]</td>
</tr>
<tr>
<td>Query Capabilities</td>
<td>OSLC servers <strong>MUST</strong> provide query capabilities to enable clients to query for resources [qm-10]</td>
</tr>
<tr>
<td>Query Syntax</td>
<td>OSLC query capabilities <strong>MUST</strong> support the OSLC Core Query Syntax and <strong>MAY</strong> use other query syntax [qm-11]</td>
</tr>
<tr>
<td>Delegated UI Dialogs</td>
<td>OSLC Services <strong>MUST</strong> offer delegated UI dialogs (creation and selections) specified via OSLC Core 3.0 Delegated Dialogs and <strong>SHOULD</strong> include discovery through a ServiceProvider resource for OSLC v2 compatibility [qm-12]</td>
</tr>
<tr>
<td>UI Preview</td>
<td>OSLC Services <strong>SHOULD</strong> offer UI previews for resources that may be referenced by other resources specified via OSLC Core 3.0 Preview and <strong>SHOULD</strong> include discovery through a server resource for OSLC v2 compatibility [qm-13]</td>
</tr>
<tr>
<td>Requirement</td>
<td>Meaning</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HTTP Basic Authentication</td>
<td>OSLC Services MAY support Basic Auth and should do so only over HTTPS [qm-14]</td>
</tr>
<tr>
<td>OAuth Authentication</td>
<td>OSLC Services SHOULD support OAuth and can indicate the required OAuth URLs via the ServiceProviderCatalog or ServiceProvider resources [qm-15]</td>
</tr>
<tr>
<td>Error Responses</td>
<td>OSLC Services SHOULD provide error responses using OSLC Core 3.0 defined error formats [qm-16]</td>
</tr>
<tr>
<td>Turtle Representations</td>
<td>OSLC services MUST provide a Turtle representation for HTTP GET requests and SHOULD support Turtle representations on POST and PUT requests. [qm-17]</td>
</tr>
<tr>
<td>RDF/XML Representations</td>
<td>OSLC services SHOULD provide an RDF/XML representation for HTTP GET requests and SHOULD support RDF/XML representations on POST and PUT requests [qm-18]</td>
</tr>
<tr>
<td>XML Representations</td>
<td>OSLC services SHOULD provide a XML representation for HTTP GET, POST and PUT requests that conform to the Core 2.0 Guidelines for XML. [qm-19]</td>
</tr>
<tr>
<td>JSON Representations</td>
<td>OSLC services MUST provide JSON-LD representations for HTTP GET, POST and PUT requests that conform to the Core Guidelines for JSON-LD [qm-20]</td>
</tr>
<tr>
<td>HTML Representations</td>
<td>OSLC services SHOULD provide HTML representations for HTTP GET requests [qm-21]</td>
</tr>
<tr>
<td>QM 1.0 response compatibility</td>
<td>OSLC QM servers MAY use application/x-oslc-qm-* media types in responses for compatibility with [OSLCQM1] [qm-22]</td>
</tr>
</tbody>
</table>

### 2.2 Specification Versioning

This specification follows the specification version guidelines given in [OSLCCore3](OSLCCore3), section 5). There are no substantive changes from [OSLC Quality Management Specification Version 2.0](OSLCQM2). All changes are all upward compatible additions and do not introduce incompatibilities with Version 2.0.

### 2.3 Namespaces

In addition to the namespace URIs and namespace prefixes `oslc`, `rdf`, `dcterms` and `foaf` defined in the [OSLCCore3], OSLC QM defines the namespace URI of `http://open-services.net/ns/qm#` with a namespace prefix of `oslc_qm`.

This specification also uses these namespace prefix definitions:

- `oslc_cm`: `http://open-services.net/ns/cm#` [OSLCCM20]
- `oslc_rm`: `http://open-services.net/ns/rm#` [OSLCRM20]

### 2.4 Resource Formats

In addition to the requirements for resource representations in [OSLCCore3], this section outlines further refinements and restrictions.

For HTTP GET requests on all OSLC QM and OSLC Core defined resource types,

- QM servers MUST provide Turtle and JSON-LD, and SHOULD provide RDF/XML and XML representations. The XML and JSON representations SHOULD follow the guidelines outlined in the [OSLC Core Representations Guidance](OSLCCore2) to maintain compatibility with [OSLCORE2]. [qm-23]
- QM clients requesting RDF/XML SHOULD be prepared for any valid RDF/XML document. QM clients requesting XML SHOULD be prepared for representations that follow the guidelines outlined in the [OSLC Core Representations Guidance](OSLCCore2).
QM servers SHOULD support an [X]HTML representation and a user interface (UI) preview as defined by UI Preview Guidance.

For HTTP PUT/POST request formats for resource type of TestPlan, TestCase, TestScript, TestExecutionRecord and TestResult:

- QM servers MUST accept Turtle and JSON-LD representations and SHOULD accept RDF/XML and XML representations. QM servers accepting RDF/XML SHOULD be prepared for any valid RDF/XML document. For XML, QM servers SHOULD be prepared for representations that follow the guidelines outlined in the OSLC Core Representations Guidance.

For HTTP GET response formats for Query requests:

- QM servers MUST provide Turtle and JSON-LD, SHOULD provide RDF/XML, XML, and MAY provide Atom Syndication Format XML representations.

When QM clients request:

- text/turtle QM servers MUST respond with Turtle representation.
- application/ld+json QM servers MUST respond with JSON-LD representation.
- application/rdf+xml QM servers SHOULD respond with RDF/XML representation without restrictions.
- application/xml QM servers SHOULD respond with OSLC-defined abbreviated XML representation as defined in the OSLC Core Representations Guidance.
- application/atom+xml QM servers MAY support Atom Syndication Format XML representation as defined in the OSLC Core Representations Guidance to maintain integrations with the legacy applications.

The Atom Syndication Format XML representation SHOULD use RDF/XML representation without restrictions for the atom:content entries representing the resource representations.

See Query Capabilities for additional information when Resource Shapes affect representation.

2.4.1 Content Negotiation

[OSLCCore3] specifies RDF representations (and specifically Turtle and JSON-LD) as a convention that all OSLC server implementations minimally provide and accept. OSLC QM server implementations are strongly encouraged to adopt this convention. Future versions of this specification are expected to require RDF representations for all operations and relax requirements for specialized XML representations.

XML Representation - identified by the application/xml content type. Format representation rules are outlined in Core OSLC Core Resource Formats section.

RDF/XML Representation - identified by the application/rdf+xml content type. No additional guidance is given.

JSON-LD Representation - identified by the application/ld+json content type. Format representation rules are specified in JSON-LD 1.0.

Atom Syndication Format XML Representation - identified by the application/atom+xml content type. Format representation rules are outlined in Core OSLC Core Resource Formats section.

2.5 Authentication

[OSLCCore3] specifies the recommended OSLC authentication mechanisms. In addition to the OSLC Core authentication requirements, OSLC QM services providers SHOULD support [OpenIDConnect].

2.6 Error Responses

[OSLCCore3] specifies the OSLC Core error responses. OSLC QM puts no additional constraints on error responses.
2.7 Pagination

OSLC QM servers **SHOULD** support pagination of query results and **MAY** support pagination of a single resource's properties as defined by [OSLCCore3]. [qm-34]

2.8 Requesting and Updating Properties

2.8.1 Requesting a Subset of Properties

A client **MAY** request a subset of a resource's properties as well as properties from a referenced resource. In order to support this behavior a server **MUST** support the `oslc.properties` and `oslc.prefix` URL parameter on a HTTP GET request on an individual resource request or a collection of resources by query. If the `oslc.properties` parameter is omitted on the request, then all resource properties **MUST** be provided in the response. [qm-35]

2.8.2 Updating a Subset of Properties

A client **MAY** request that a subset of a resource's properties be updated by using the [LDPPatch] PATCH method. [qm-36]

For compatibility with [OSLCCore2], QM servers **MAY** also support partial update by identifying those properties to be modified using the `oslc.properties` URL parameter on a HTTP PUT request. [qm-37]

If the parameter `oslc.properties` contains a valid resource property on the request that is not provided in the content, the server **MUST** set the resource's property to a null or empty value. If the parameter `oslc.properties` contains an invalid resource property, then a 409 Conflict **MUST** be returned. [qm-38]

2.8.3 Updating Multi-Valued Properties

For multi-valued properties that contain a large number of values, it may be difficult and inefficient to add or remove property values. OSLC QM servers **SHOULD** provide support for a partial update of the multi-valued properties as defined by OSLC Core Partial Update. [qm-39]

2.9 Labels for Relationships

*This section is non-normative.*

QM resource relationships to other resources are represented by RDF properties. Instances of a relationship - often called links - are RDF triples with a subject URI, a predicate that is the property, and a value (or object) that is the URI of target resource. When a link is to be presented in a user interface, it may be helpful to display an informative and useful textual label instead of or in addition to the URI of the predicate and or object. There are three items that clients could display:

- **The property:** OSLC recommends using the rdfs:label property of the rdf:Property from the vocabulary to display the property.
- **The value, or object of the triple:** OSLC recommends using OSLC resource preview [OSLCPreview] to obtain an appropriate icon and label, and possibly a small and/or large dialog for displaying the object.
- **The link:** The link is a combination of the subject, predicate and object of the triple (RDF statement or assertion). In the case where the link requires a unique label that is not available from the target resource, only then OSLC servers may support a dcterms:title on a reified statement to provide a label for a link that describes the assertion itself.

Turtle example using a reified statement:

```
Example 1
@prefix ns0: <http://open-services.net/ns/qm#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
<http://example.com/tests/4321>
```
a <http://open-services.net/ns/qm#TestCase> ;
  ns0:uses <http://anotherexample.com/testscript/123> .

<http://njh.me/#link1>
a rdf:Statement ;
rdf:subject <http://example.com//4321> ;
rdf:_predicate ns0:uses ;
rdf:object <http://anotherexample.com/testscript/123> ;
dcterms:title "Test Script 123: " .

JSON-LD example using reified statement:

Example 2
{
  "@context": {
    "dcterms": "http://purl.org/dc/terms/",
    "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
    "oslc": "http://open-services.net/ns/core#",
    "oslc_qm": "http://open-services.net/ns/qm#"
  },
  "@id": "http://example.com/testscripts/4321",
  "@type": "oslc_qm:TestScript",
  "oslc_cm:relatedChangeRequest": {
    "@id": "http://anotherexample.com/defects/123",
    "dcterms:title": "Defect 123: Problems during install"
  }
}
3. Vocabulary Terms and Constraints

OSLC QM Resources 2.1 Defines the vocabulary terms and constraints for OSLC Quality Management resources. These terms and constraints are specified according to [OSLCCoreVocab].

The intent is to define resources needed to support common integration scenarios and not to provide a comprehensive definition of QM resources like Test Case. The resource formats may not match exactly the native models supported by quality management service providers, but are intended to be compatible with them. The approach to supporting these scenarios is to delegate operations, as driven by service provider contributed user interfaces, as much as possible and not require a service provider to expose its complete data model and application logic.
4. QM Server Capabilities

4.1 Resource Shapes

OSLC QM servers SHOULD support Resource Shapes as defined in [OSLCSHapes]. [qm-40]

4.2 Service Provider Resources

OSLC QM servers MUST support OSLC Discovery capabilities defined by [OSLCCore3]. [qm-41]

OSLC domain-name servers MAY provide a ServiceProvider Resource that can be retrieved at a implementation dependent URI. [qm-42]

OSLC QM servers MAY provide a ServiceProviderCatalog Resource that can be retrieved at a implementation dependent URI. [qm-43]

OSLC QM servers MAY provide a oslc:serviceProvider property for their defined resources that will be the URI to a ServiceProvider Resource. [qm-44]

OSLC QM servers MUST supply a value of http://open-services.net/ns/qm# for the property oslc:domain on either oslc:Service or oslc:ServiceProviderCatalog resources. [qm-45]

4.3 Creation Factories

OSLC QM servers MUST support [OSLCCore3] Creation Factories and list them in the Service Provider Resource as defined by OSLC Core. OSLC QM servers SHOULD support Resource Shapes for Creation Factories as defined in [OSLCSHapes]. [qm-46]

4.4 Query Capabilities

OSLC QM servers SHOULD support the Query Capabilities as defined by [OSLCCore3]. OSLC QM servers SHOULD support Resource Shapes for Query Capability as defined in [OSLCSHapes]. [qm-47]

The Query Capability, if supported, MUST support these parameters: [qm-48]

- oslc.where [qm-49]
- oslc.select [qm-50]
- oslc.properties [qm-51]
- oslc.prefix [qm-52]

If shape information is NOT present with the Query Capability, servers SHOULD use these default properties to contain the result: [qm-53]

- For RDF/XML and XML, use rdf:Description and rdfs:member as defined in OSLC Core RDF/XML Examples.
- For JSON, the query results are contained within oslc:results array. See OSLC Core Representation Guidance for JSON.

4.5 Delegated Uls

OSLC QM servers MUST support the selection and creation of resources by delegated web-based user interface delegated dialogs Delegated as defined by [OSLCCore3]. [qm-54]

OSLC QM servers MAY support the pre-filling of creation dialogs based on the definition [OSLCCore3]. [qm-55]
### 5. Conformance

Implementations of this specification need to satisfy the following conformance clauses.

<table>
<thead>
<tr>
<th>Clause Number</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>qm-1</td>
<td>This specification is based on [OSLCCore3]. OSLC QM servers MUST be compliant with both the core specification. MUST follow all the mandatory requirements in the normative sections of this specification, and SHOULD follow all the guidelines and recommendations in both these specifications.</td>
</tr>
<tr>
<td>qm-2</td>
<td>An OSLC QM server MUST implement the domain vocabulary defined in OSLC Quality Management Version 2.1. Part 2: Vocabulary</td>
</tr>
<tr>
<td>qm-3</td>
<td>OSLC services MAY Ignore unknown content and OSLC clients MUST preserve unknown content</td>
</tr>
<tr>
<td>qm-4</td>
<td>OSLC service MUST support resource operations via standard HTTP operations</td>
</tr>
<tr>
<td>qm-5</td>
<td>OSLC services MAY provide paging for resources but only when specifically requested by client</td>
</tr>
<tr>
<td>qm-6</td>
<td>OSLC services MUST support request for a subset of a resource’s properties via the oslc.properties URL parameter retrieval via HTTP GET</td>
</tr>
<tr>
<td>qm-7</td>
<td>OSLC services MAY support partial update of resources using patch semantics and MAY support via HTTP PUT</td>
</tr>
<tr>
<td>qm-8</td>
<td>OSLC services MUST support OSLC Core 3.0 Discovery, MAY provide a ServiceProviderCatalog and SHOULD provide a ServiceProvider resource for Core v2 compatibility.</td>
</tr>
<tr>
<td>qm-9</td>
<td>OSLC servers MUST provide LDPC creation factories to enable resource creation of Quality Management resources via HTTP POST</td>
</tr>
<tr>
<td>qm-10</td>
<td>OSLC servers MUST provide query capabilities to enable clients to query for resources</td>
</tr>
<tr>
<td>qm-11</td>
<td>OSLC query capabilities MUST support the OSLC Core Query Syntax and MAY use other query syntax</td>
</tr>
<tr>
<td>qm-12</td>
<td>OSLC Services MUST offer delegated UI dialogs (creation and selections) specified via OSLC Core 3.0 Delegated Dialogs and SHOULD include discovery through a ServiceProvider resource for OSLC v2 compatibility</td>
</tr>
<tr>
<td>qm-13</td>
<td>OSLC Services SHOULD offer UI previews for resources that may be referenced by other resources specified via OSLC Core 3.0 Preview and SHOULD include discovery through a server resource for OSLC v2 compatibility</td>
</tr>
<tr>
<td>qm-14</td>
<td>OSLC Services MAY support Basic Auth and should do so only over HTTPS</td>
</tr>
<tr>
<td>qm-15</td>
<td>OSLC Services SHOULD support OAuth and can indicate the required OAuth URLs via the ServiceProviderCatalog or ServiceProvider resources</td>
</tr>
<tr>
<td>qm-16</td>
<td>OSLC Services SHOULD provide error responses using OSLC Core 3.0 defined error formats</td>
</tr>
<tr>
<td>qm-17</td>
<td>OSLC services MUST provide a Turtle representation for HTTP GET requests and SHOULD support Turtle representations on POST and PUT requests.</td>
</tr>
<tr>
<td>qm-18</td>
<td>OSLC services SHOULD provide an RDF/XML representation for HTTP GET requests and SHOULD support RDF/XML representations on POST and PUT requests</td>
</tr>
<tr>
<td>qm-19</td>
<td>OSLC services SHOULD provide a XML representation for HTTP GET, POST and PUT requests that conform to the Core 2.0 Guidelines for XML.</td>
</tr>
<tr>
<td>qm-20</td>
<td>OSLC services MUST provide JSON-LD representations for HTTP GET, POST and PUT requests that conform to the Core Guidelines for JSON-LD</td>
</tr>
<tr>
<td>qm-21</td>
<td>OSLC services SHOULD provide HTML representations for HTTP GET requests</td>
</tr>
<tr>
<td>qm-22</td>
<td>OSLC QM servers MAY use application/x-oslc-qm-* media types in responses for compatibility with [OSLCCQM1]</td>
</tr>
<tr>
<td>qm-23</td>
<td>QM servers MUST provide Turtle and JSON-LD, and SHOULD provide RDF/XML and XML representations. The XML and JSON representations SHOULD follow the guidelines outlined in the OSLC Core Representations Guidance to maintain compatibility with [OSLCCore2].</td>
</tr>
<tr>
<td>Clause Number</td>
<td>Requirement</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>qm-24</td>
<td>QM clients requesting RDF/XML SHOULD be prepared for any valid RDF/XML document. QM clients requesting XML SHOULD be prepared for representations that follow the guidelines outlined in the OSLC Core Representations Guidance.</td>
</tr>
<tr>
<td>qm-25</td>
<td>QM servers SHOULD support an [X]HTML representation and a user interface (UI) preview as defined by UI Preview Guidance.</td>
</tr>
<tr>
<td>qm-26</td>
<td>QM servers MUST accept Turtle and JSON-LD representations and SHOULD accept RDF/XML and XML representations. QM servers accepting RDF/XML SHOULD be prepared for any valid RDF/XML document. For XML, QM servers SHOULD be prepared for representations that follow the guidelines outlined in the OSLC Core Representations Guidance.</td>
</tr>
<tr>
<td>qm-27</td>
<td>QM servers MUST provide Turtle and JSON-LD, SHOULD provide RDF/XML, XML, and MAY provide Atom Syndication Format XML representations.</td>
</tr>
<tr>
<td>qm-28</td>
<td>text/turtle QM servers MUST respond with Turtle representation.</td>
</tr>
<tr>
<td>qm-29</td>
<td>application/ld+json QM servers MUST respond with JSON-LD representation.</td>
</tr>
<tr>
<td>qm-30</td>
<td>application/rdf+xml QM servers SHOULD respond with RDF/XML representation without restrictions.</td>
</tr>
<tr>
<td>qm-31</td>
<td>application/xml QM servers SHOULD respond with OSLC-defined abbreviated XML representation as defined in the OSLC Core Representations Guidance.</td>
</tr>
<tr>
<td>qm-32</td>
<td>application/atom+xml QM servers MAY support Atom Syndication Format XML representation as defined in the OSLC Core Representations Guidance to maintain integrations with the legacy applications.</td>
</tr>
<tr>
<td>qm-33</td>
<td>[OSLCCore3] specifies the recommended OSLC authentication mechanisms. In addition to the OSLC Core authentication requirements, OSLC QM services providers SHOULD support [OpenIDConnect].</td>
</tr>
<tr>
<td>qm-34</td>
<td>OSLC QM servers SHOULD support pagination of query results and MAY support pagination of a single resource's properties as defined by [OSLCCore3].</td>
</tr>
<tr>
<td>qm-35</td>
<td>A client MAY request a subset of a resource's properties as well as properties from a referenced resource. In order to support this behavior a server MUST support the oslc.properties and oslc.prefix URL parameter on a HTTP GET request on individual resource request or a collection of resources by query. If the oslc.properties parameter is omitted on the request, then all resource properties MUST be provided in the response.</td>
</tr>
<tr>
<td>qm-36</td>
<td>A client MAY request that a subset of a resource's properties be updated by using the [LDPPatch] PATCH method.</td>
</tr>
<tr>
<td>qm-37</td>
<td>For compatibility with [OSLCCore2], QM servers MAY also support partial update by identifying those properties to be modified using the oslc.properties URL parameter on a HTTP PUT request.</td>
</tr>
<tr>
<td>qm-38</td>
<td>If the parameter oslc.properties contains a valid resource property on the request that is not provided in the content, the server MUST set the resource's property to a null or empty value. If the parameter oslc.properties contains an invalid resource property, then a 409 Conflict MUST be returned.</td>
</tr>
<tr>
<td>qm-39</td>
<td>For multi-valued properties that contain a large number of values, it may be difficult and inefficient to add or remove property values. OSLC QM servers SHOULD provide support for a partial update of the multi-valued properties as defined by OSLC Core Partial Update.</td>
</tr>
<tr>
<td>qm-40</td>
<td>OSLC QM servers SHOULD support Resource Shapes as defined in [OSLCShapes].</td>
</tr>
<tr>
<td>qm-41</td>
<td>OSLC QM servers MUST support OSLC Discovery capabilities defined by [OSLCCore3].</td>
</tr>
<tr>
<td>qm-42</td>
<td>OSLC domain-name servers MAY provide a ServiceProvider Resource that can be retrieved at a implementation dependent URI.</td>
</tr>
<tr>
<td>qm-43</td>
<td>OSLC QM servers MAY provide a ServiceProviderCatalog Resource that can be retrieved at a implementation dependent URI.</td>
</tr>
<tr>
<td>qm-44</td>
<td>OSLC QM servers MAY provide a oslc:serviceProvid er property for their defined resources that will be the URI to a ServiceProvider Resource.</td>
</tr>
<tr>
<td>qm-45</td>
<td>OSLC QM servers MUST Supply a value of <a href="http://open-services.net/ns/qm#">http://open-services.net/ns/qm#</a> for the property oslc:domain on either oslc:Service of oslc:ServiceProviderCatalog resources.</td>
</tr>
</tbody>
</table>
OSLC QM servers **MUST** support [OSLCCore3] Creation Factories and list them in the Service Provider Resource as defined by OSLC Core. OSLC QM servers **SHOULD** support Resource Shapes for Creation Factories as defined in [OSLCShapes].

OSLC QM servers **SHOULD** support the Query Capabilities as defined by [OSLCCore3]. OSLC QM servers **SHOULD** support Resource Shapes for Query Capability as defined in [OSLCShapes].

The Query Capability, if supported, **MUST** support these parameters:

<table>
<thead>
<tr>
<th>Clause Number</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>qm-46</td>
<td>oslc.where</td>
</tr>
<tr>
<td>qm-50</td>
<td>oslc.select</td>
</tr>
<tr>
<td>qm-51</td>
<td>oslc.properties</td>
</tr>
<tr>
<td>qm-52</td>
<td>oslc.prefix</td>
</tr>
</tbody>
</table>

If shape information is NOT present with the Query Capability, servers **SHOULD** use these default properties to contain the result:

OSLC QM servers **MUST** support the selection and creation of resources by delegated web-based user interface delegated dialogs Delegated as defined by [OSLCCore3].

OSLC QM servers **MAY** support the pre-filling of creation dialogs based on the definition [OSLCCore3].

<table>
<thead>
<tr>
<th>Clause Number</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>qm-47</td>
<td>oslc.detail</td>
</tr>
<tr>
<td>qm-54</td>
<td>oslc.properties</td>
</tr>
<tr>
<td>qm-55</td>
<td>oslc.prefix</td>
</tr>
</tbody>
</table>
## Appendix A. Version Compatibility with 2.0 Specifications

The following differences were noted when migrating the open-services.net version 2.0 Quality Management specification to the version 2.1 OASIS template based upon Change Management

<table>
<thead>
<tr>
<th>Item</th>
<th>V2 Status</th>
<th>V3 Status</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Service Provider Compliance requirements are listed</td>
<td>These requirements appear to be hosted under the heading of Discovery</td>
<td>No action</td>
</tr>
<tr>
<td>03</td>
<td>Query MUST scope limited to oslc.where and oslc.select</td>
<td>Query MUST scope extended oslc.properties and oslc.prefix</td>
<td>No action</td>
</tr>
<tr>
<td>04</td>
<td>V1 compatibility included</td>
<td>V1 compatibility not included</td>
<td>No action</td>
</tr>
</tbody>
</table>
Appendix B. Acknowledgements

This section is non-normative.

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

**Project Governing Board:**
- James Amsden, IBM (co-chair)
- Andrii Berezovskiy, KTH (co-chair)
- Bill Chown, Mentor
- Wesley Coelho, Tasktop
- Axel Reichwein, Koneksys

**Technical Steering Committee:**
- James Amsden, IBM
- Andrii Berezovskiy, KTH
- Axel Reichwein, Koneksys

**OSLC QM 2.0 contributors:**
- Gray Bachelor (IBM - Editor)
- Dave Johnson (IBM)
- Ingrid Jorgensen (Tieto)
- Michael Pena (Sogeti)
- Nigel Lawrence (IBM)
- Paul McMahan (IBM, OSLC-QM Lead)
- Scott Bosworth (IBM)
- Scott Fairbrother (IBM)