



OSLC Quality Management Version 2.1. Part 1: Specification

Project Specification 01 27 August 2020

This stage:

<https://docs.oasis-open-projects.org/oslc-op/qm/v2.1/ps01/quality-management-spec.html>
(Authoritative)
<https://docs.oasis-open-projects.org/oslc-op/qm/v2.1/ps01/quality-management-spec.pdf>

Previous 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>

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](#)

Andrii Berezovskyi (andriib@kth.se), [KTH](#)

Gray Bachelor (gray_bachelor@uk.ibm.com), [IBM](#)

Additional components:

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

- OSLC Quality Management Version 2.1. Part 1: Specification (this document). [quality-management-spec.html](#)
- OSLC Quality Management Version 2.1. Part 2: Vocabulary. [quality-management-vocab.html](#)
- OSLC Quality Management Version 2.1. Part 3: Constraints. [quality-management-shapes.html](#)
- OSLC Quality Management Version 2.1. Part 4: Machine Readable Vocabulary Terms. [quality-management-vocab.ttl](#)
- OSLC Quality Management Version 2.1. Part 5: Machine Readable Constraints. [quality-management-shapes.ttl](#)

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/specifications/>.

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]

OSLC Quality Management Version 2.1. Part 1: Specification. Edited by Jim Amsden, Andrii Berezovskyi, and Gray Bachelor. 27 August 2020. OASIS Project Specification 01.

<https://docs.oasis-open-projects.org/oslc-op/qm/v2.1/ps01/quality-management-spec.html>. Latest stage: <https://docs.oasis-open-projects.org/oslc-op/qm/v2.1/quality-management-spec.html>.

Notices

Copyright © OASIS Open 2020. 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 specification is published under the [Attribution 4.0 International \(CC BY 4.0\)](#). Portions of this specification are also provided under the [Apache License 2.0](#).

All contributions made to this project have been made under the [OASIS Contributor License Agreement \(CLA\)](#).

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 [Open Projects IPR Statements page](#).

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 Open Project or 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 Project Specification or OASIS Standard, to notify the 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 Open Project 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 Open Project 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](https://www.oasis-open.org/), 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](#)
 - 1.1 [Overview](#)
 - 1.2 [Terminology](#)
 - 1.3 [References](#)
 - 1.3.1 [Normative references](#)
 - 1.3.2 [Informative references](#)
- 2. [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. Change History](#)
- [Appendix C. 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 as 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](https://archive.open-services.net/bin/view/Main/OslcCoreSpecification). Finalized. URL: <https://archive.open-services.net/bin/view/Main/OslcCoreSpecification>

[OSLCCore3]

Jim Amsden; S. Speicher. [OSLC Core 3.0](https://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd03/part1-overview/oslc-core-v3.0-csprd03-part1-overview.html). Committee Specification Public Review Draft. URL: <https://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd03/part1-overview/oslc-core-v3.0-csprd03-part1-overview.html>

[OSLCCoreVocab]

Jim Amsden; S. Padgett; S. Speicher. [*OSLC Core Vocabulary*](#). Committee Specification Public Review Draft. URL: <https://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd03/part7-core-vocabulary/oslc-core-v3.0-csprd03-part7-core-vocabulary.html>

[OSLCPreview]

[*OSLC Resource Preview 3.0*](#). Working Draft. URL: <https://oslc-op.github.io/oslc-specs/specs/core/resource-preview.html>

[OSLCQM1]

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

[OSLCShapes]

Arthur Ryman; Jim Amsden. [*OSLC Resource Shape 3.0*](#). Committee Specification Public Review Draft. URL: <https://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd03/part6-resource-shape/oslc-core-v3.0-csprd03-part6-resource-shape.html>

[OpenIDConnect]

[*OpenID Connect*](#). URL: <http://openid.net/connect/>

1.3.2 Informative references

[HTTP11]

R. Fielding, Ed.; J. Reschke, Ed.. [*Hypertext Transfer Protocol \(HTTP/1.1\): Message Syntax and Routing*](#). June 2014. Proposed Standard. URL: <https://httpwg.org/specs/rfc7230.html>

[LDP]

Steve Speicher; John Arwe; Ashok Malhotra. [*Linked Data Platform 1.0*](#). 26 February 2015. W3C Recommendation. URL: <https://www.w3.org/TR/ldp/>

[LDPPatch]

[*Linked Data Patch Format*](#). Working Group Note. URL: <http://www.w3.org/TR/ldpatch/>

[OSLCCM20]

Steve Speicher. [*Open Services for Lifecycle Collaboration Change Management Specification Version 2.0*](#). Finalized. URL: <https://archive.open-services.net/bin/view/Main/CmSpecificationV2>

[OSLCQM2]

Paul McMahan. [*OSLC Quality Management Specification Version 2.0*](https://archive.open-services.net/bin/view/Main/QmSpecificationV2.html). Finalized. URL: <https://archive.open-services.net/bin/view/Main/QmSpecificationV2.html>

[OSLCRM20]

Ian Green. [*Open Services for Lifecycle Collaboration Requirements Management Specification Version 2.0*](https://archive.open-services.net/bin/view/Main/RmSpecificationV2). Finalized. URL: <https://archive.open-services.net/bin/view/Main/RmSpecificationV2>

[WEBARCH]

Ian Jacobs; Norman Walsh. [*Architecture of the World Wide Web, Volume One*](https://www.w3.org/TR/webarch/). 15 December 2004. W3C Recommendation. URL: <https://www.w3.org/TR/webarch/>

2. 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.

Requirement	Meaning
Unknown properties and content	OSLC services MAY ignore unknown content and OSLC clients MUST preserve unknown content [qm-3]
Resource Operations	OSLC service MUST support resource operations via standard HTTP operations [qm-4]
Resource Paging	OSLC services MAY provide paging for resources but only when specifically requested by client [qm-5]
Partial Resource Representations	OSLC services MUST support request for a subset of a resource's properties via the oslc.properties URL parameter retrieval via HTTP GET [qm-6]
Partial Update	OSLC services MAY support partial update of resources using patch semantics and MAY support via HTTP PUT [qm-7]
Discovery	OSLC servers MUST support OSLC Core 3.0 Discovery, MAY provide a ServiceProviderCatalog and SHOULD provide a ServiceProvider resource for Core v2 compatibility. [qm-8]
Creation Factories	OSLC servers MUST provide LDPC creation factories to enable resource creation of Quality Management resources via HTTP POST [qm-9]

Requirement	Meaning
Query Capabilities	OSLC servers MUST provide query capabilities to enable clients to query for resources [qm-10]
Query Syntax	OSLC query capabilities MUST support the OSLC Core Query Syntax and MAY use other query syntax [qm-11]
Delegated UI Dialogs	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 [qm-12]
UI Preview	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 [qm-13]
HTTP Basic Authentication	OSLC Services MAY support Basic Auth and should do so only over HTTPS [qm-14]
OAuth Authentication	OSLC Services SHOULD support OAuth and can indicate the required OAuth URLs via the ServiceProviderCatalog or ServiceProvider resources [qm-15]
Error Responses	OSLC Services SHOULD provide error responses using OSLC Core 3.0 defined error formats [qm-16]
Turtle Representations	OSLC services MUST provide a Turtle representation for HTTP GET requests and SHOULD support Turtle representations on POST and PUT requests. [qm-17]
RDF/XML Representations	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]
XML Representations	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]
JSON Representations	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]
HTML Representations	OSLC services SHOULD provide HTML representations for HTTP GET requests [qm-21]
QM 1.0 response compatibility	OSLC QM servers MAY use <code>application/x-oslq-qm-*</code> media types in responses for compatibility with [OSLCQM1] [qm-22]

2.2 Specification Versioning

This specification follows the specification version guidelines given in [OSLC Core Version 3.0. Part 1: Overview](#) ([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](#) to maintain compatibility with [OSLCCore2]. [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](#). [qm-24]
- QM servers **SHOULD** support an [X]HTML representation and a user interface (UI) preview as defined by [UI Preview Guidance](#) [qm-25]

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](#). [qm-26]

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. [qm-27]

When QM clients request:

- **text/turtle** QM servers **MUST** respond with Turtle representation. [qm-28]

- `application/ld+json` QM servers **MUST** respond with JSON-LD representation. [qm-29]
- `application/rdf+xml` QM servers **SHOULD** respond with RDF/XML representation without restrictions. [qm-30]
- `application/xml` QM servers **SHOULD** respond with OSLC-defined abbreviated XML representation as defined in the [OSLC Core Representations Guidance](#) [qm-31]
- `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. [qm-32]
- The Atom Syndication Format XML representation **SHOULD** use RDF/XML representation without restrictions for the `atom:content` entries representing the resource representations. [qm-33]

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]. [qm-34]

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 [\[OSLCCore3\]](#). [\[qm-35\]](#)

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 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. [\[qm-36\]](#)

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. [\[qm-37\]](#)

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. [\[qm-38\]](#)

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. [\[qm-39\]](#)

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](#). [\[qm-40\]](#)

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-41]

4.2 Service Provider Resources

OSLC QM servers **MUST** support OSLC Discovery capabilities defined by [OSLCCore3]. [qm-42]

OSLC domain-name servers **MAY** provide a ServiceProvider Resource that can be retrieved at a implementation dependent URI. [qm-43]

OSLC QM servers **MAY** provide a ServiceProviderCatalog Resource that can be retrieved at a implementation dependent URI. [qm-44]

OSLC QM servers **MAY** provide a `oslc:serviceProvider` property for their defined resources that will be the URI to a ServiceProvider Resource. [qm-45]

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-46]

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-47]

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-48]

The Query Capability, if supported, **MUST** support these parameters: [qm-49]

- `oslc.where` [qm-50]
- `oslc.select` [qm-51]
- `oslc.properties` [qm-52]
- `oslc.prefix` [qm-53]

If shape information is NOT present with the Query Capability, servers **SHOULD** use these default

properties to contain the result:

- 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](#)

[qm-54]

4.5 Delegated UIs

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-55]

OSLC QM servers **MAY** support the pre-filling of creation dialogs based on the definition [OSLCCore3]. [qm-56]

5. Conformance

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

Clause Number	Requirement
qm-1	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-2	An OSLC QM server MUST implement the domain vocabulary defined in OSLC Quality Management Version 2.1. Part 2: Vocabulary
qm-3	OSLC services MAY ignore unknown content and OSLC clients MUST preserve unknown content
qm-4	OSLC service MUST support resource operations via standard HTTP operations
qm-5	OSLC services MAY provide paging for resources but only when specifically requested by client
qm-6	OSLC services MUST support request for a subset of a resource's properties via the oslc.properties URL parameter retrieval via HTTP GET
qm-7	OSLC services MAY support partial update of resources using patch semantics and MAY support via HTTP PUT
qm-8	OSLC servers MUST support OSLC Core 3.0 Discovery, MAY provide a ServiceProviderCatalog and SHOULD provide a ServiceProvider resource for Core v2 compatibility.
qm-9	OSLC servers MUST provide LDPC creation factories to enable resource creation of Quality Management resources via HTTP POST
qm-10	OSLC servers MUST provide query capabilities to enable clients to query for resources
qm-11	OSLC query capabilities MUST support the OSLC Core Query Syntax and MAY use other query syntax
qm-12	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
qm-13	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
qm-14	OSLC Services MAY support Basic Auth and should do so only over HTTPS

Clause Number	Requirement
qm-15	OSLC Services SHOULD support OAuth and can indicate the required OAuth URLs via the ServiceProviderCatalog or ServiceProvider resources
qm-16	OSLC Services SHOULD provide error responses using OSLC Core 3.0 defined error formats
qm-17	OSLC services MUST provide a Turtle representation for HTTP GET requests and SHOULD support Turtle representations on POST and PUT requests.
qm-18	OSLC services SHOULD provide an RDF/XML representation for HTTP GET requests and SHOULD support RDF/XML representations on POST and PUT requests
qm-19	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-20	OSLC services MUST provide JSON-LD representations for HTTP GET, POST and PUT requests that conform to the Core Guidelines for JSON-LD
qm-21	OSLC services SHOULD provide HTML representations for HTTP GET requests
qm-22	OSLC QM servers MAY use <code>application/x-oslq-qm-*</code> media types in responses for compatibility with [OSLCQM1]
qm-23	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].
qm-24	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 .
qm-25	QM servers SHOULD support an [X]HTML representation and a user interface (UI) preview as defined by UI Preview Guidance
qm-26	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 .
qm-27	QM servers MUST provide Turtle and JSON-LD, SHOULD provide RDF/XML, XML, and MAY provide Atom Syndication Format XML representations.
qm-28	<code>text/turtle</code> QM servers MUST respond with Turtle representation.
qm-29	<code>application/ld+json</code> QM servers MUST respond with JSON-LD representation.
qm-30	<code>application/rdf+xml</code> QM servers SHOULD respond with RDF/XML representation without restrictions.

Clause Number	Requirement
qm-31	<code>application/xml</code> QM servers SHOULD respond with OSLC-defined abbreviated XML representation as defined in the OSLC Core Representations Guidance
qm-32	<code>application/atom+xml</code> QM servers MAY support Atom Syndication Format XML representation as defined in the OSLC Core Representations Guidance to maintain integrations with the legacy applications.
qm-33	The Atom Syndication Format XML representation SHOULD use RDF/XML representation without restrictions for the <code>atom:content</code> entries representing the resource representations.
qm-34	[OSLCCore3] specifies the recommended OSLC authentication mechanisms. In addition to the OSLC Core authentication requirements, OSLC QM services providers SHOULD support [OpenIDConnect].
qm-35	OSLC QM servers SHOULD support pagination of query results and MAY support pagination of a single resource's properties as defined by [OSLCCore3].
qm-36	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 <code>oslc.properties</code> and <code>oslc.prefix</code> URL parameter on a HTTP GET request on individual resource request or a collection of resources by query. If the <code>oslc.properties</code> parameter is omitted on the request, then all resource properties MUST be provided in the response.
qm-37	A client MAY request that a subset of a resource's properties be updated by using the [LDPPatch] <code>PATCH</code> method.
qm-38	For compatibility with [OSLCCore2], QM servers MAY also support partial update by identifying those properties to be modified using the <code>oslc.properties</code> URL parameter on a HTTP PUT request.
qm-39	If the parameter <code>oslc.properties</code> 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 <code>oslc.properties</code> contains an invalid resource property, then a 409 Conflict MUST be returned.
qm-40	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-41	OSLC QM servers SHOULD support Resource Shapes as defined in [OSLCShapes].
qm-42	OSLC QM servers MUST support OSLC Discovery capabilities defined by [OSLCCore3].
qm-43	OSLC domain-name servers MAY provide a ServiceProvider Resource that can be retrieved at a implementation dependent URI.

Clause Number	Requirement
qm-44	OSLC QM servers MAY provide a ServiceProviderCatalog Resource that can be retrieved at a implementation dependent URI.
qm-45	OSLC QM servers MAY provide a oslc:serviceProvider property for their defined resources that will be the URI to a ServiceProvider Resource.
qm-46	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-47	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-48	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-49	The Query Capability, if supported, MUST support these parameters:
qm-50	oslc.where
qm-51	oslc.select
qm-52	oslc.properties
qm-53	oslc.prefix
qm-54	<p>If shape information is NOT present with the Query Capability, servers SHOULD use these default properties to contain the result:</p> <ul style="list-style-type: none"> • 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
qm-55	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-56	OSLC QM servers MAY support the pre-filling of creation dialogs based on the definition [OSLCCore3].

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

Item	V2 status	V3 status	Remark
01	Service Provider Compliance requirements are listed	These requirements appear to be hosted under the heading of Discovery	No action
03	Query MUST scope limited to osc.where and osc.select	Query MUST scope extended osc.properties and osc.prefix	No action
04	V1 compatibility included	V1 compatibility not included	No action

Appendix B. Change History

This section is non-normative.

Revision	Date	Editor	Changes Made
CSPRD01 (not published)	2018-08-08	Jim Amsden and Gray Bachelor	<ul style="list-style-type: none">• Initial CSPRD01 for V2.1
PSD02	2019-11-14	Andrii Berezovskyi	<ul style="list-style-type: none">• Migration to the OASIS OSLC Open Project• Publication of the vocabulary and shapes
PS01	2020-08-27	Andrii Berezovskyi	<ul style="list-style-type: none">• Split vocabulary and shapes parts as well as reference machine-readable definitions as normative parts of the spec• Change shapes base URI• Add shapes metadata

Appendix C. 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 Berezovskyi, KTH (co-chair)
Bill Chown, Mentor
Wesley Coelho, Tasktop
Axel Reichwein, Koneksys

Technical Steering Committee:

James Amsden, IBM
Andrii Berezovskyi, 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)