



OSLC Requirements Management Version 2.1. Part 1: Specification

Project Specification Draft 02 01 October 2019

This stage:

<https://docs.oasis-open-projects.org/oslc-op/rm/v2.1/psd02/requirements-management-spec.html> (Authoritative)
<https://docs.oasis-open-projects.org/oslc-op/rm/v2.1/psd02/requirements-management-spec.pdf>

Previous stage:

<http://docs.oasis-open.org/oslc-domains/oslc-rm/v2.1/cs01/part1-requirements-management-spec/oslc-rm-v2.1-cs01-part1-requirements-management-spec.html> (Authoritative)
<http://docs.oasis-open.org/oslc-domains/oslc-rm/v2.1/cs01/part1-requirements-management-spec/oslc-rm-v2.1-cs01-part1-requirements-management-spec.pdf>

Latest stage:

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

Latest version:

<https://open-services.net/spec/rm/latest>

Latest editor's draft:

<https://open-services.net/spec/rm/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:

Mark Schulte (mark.d.schulte@boeing.com), [The Boeing Company](#)
Jad El-khoury (jad@kth.se), [KTH The Royal Institute of Technology](#)

Additional components:

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

- *OSLC Requirements Management Version 2.1. Part 1: Specification (this document)*. [requirements-management-spec.html](#)

Standards Track Work Product

- *OSLC Requirements Management Version 2.1. Part 2: Vocabulary*. [requirements-management-vocab.html](#)

Related work:

This specification is related to:

- *Open Services for Lifecycle Collaboration Requirements Management Specification Version 2.0*. <http://open-services.net/bin/view/Main/RmSpecificationV2>

RDF Namespaces:

<http://open-services.net/ns/rm#>

Abstract:

This specification defines the OSLC Requirements Management domain. The specification supports key RESTful web service interfaces for the management of Requirements, Requirements Collections and supporting resources defined in the OSLC Core specification. 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-RM-2.1-Part1]

OSLC Requirements Management Version 2.1. Part 1: Specification. Edited by Mark Schulte and Jad El-khoury. 01 October 2019. OASIS Project Specification Draft 02. <https://docs.oasis-open-projects.org/oslc-op/rm/v2.1/psd02/requirements-management-spec.html>. Latest stage: <https://docs.oasis-open-projects.org/oslc-op/rm/v2.1/requirements-management-spec.html>.

Notices

Copyright © OASIS Open 2019. 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](#), 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 [Terminology](#)
 - 1.2 [References](#)
 - 1.3 [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](#)
 - 2.5 [Authentication](#)
 - 2.6 [Error Responses](#)
 - 2.7 [Pagination](#)
 - 2.8 [Requesting and Updating Properties](#)
 - 2.9 [Labels for Relationships](#)
- 3. [Vocabulary Terms and Constraints](#)
- 4. [RM Server Capabilities](#)
 - 4.1 [Server Resources](#)
 - 4.2 [Creation Factories](#)
 - 4.3 [Query Capabilities](#)
 - 4.4 [Delegated Uls](#)
 - 4.5 [Usage Identifiers](#)
- 5. [Conformance](#)
- Appendix A. [Version Compatibility](#)
 - A.1 [Version Compatibility with 2.0 Specifications](#)
 - A.2 [Version Compatibility with 1.0 Specifications](#)
- Appendix B. [Acknowledgements](#)
- Appendix C. [Change History](#)

1. Introduction

This section is non-normative.

This specification defines the OSLC Requirements Management domain, also known as OSLC RM. The specification supports key RESTful web service interfaces for software Requirements Management systems. OSLC takes an open, loosely coupled approach to specific lifecycle integration scenarios. The scenarios and this specification were created by the OASIS OSLC OP TSC.

This specification builds on the Open Services for Lifecycle Collaboration Core Specification [OSLCCore3] to define the resources, properties and operations supported by an OSLC Requirements Definition and Management (OSLC-RM) server.

Requirements Management resources include Requirements, Requirements Collections and supporting resources defined in the OSLC Core specification. The properties defined describe these resources and the relationships between resources. Operations are defined in terms of HTTP methods and MIME type handling. The resources, properties and operations defined do not form a comprehensive interface to Requirements Definition and Management, but instead target specific integration use cases documented by the OASIS OSLC OP.

1.1 Terminology

This section is non-normative.

Terminology uses and extends the terminology and capabilities of [OSLCCore3].

Requirement Resource

Requirements are the basis for defining what the system stakeholders (users, customers, suppliers and so on) need from a system and also what the system must do in order to meet those needs, and how the surrounding processes must be orchestrated so that quality, scope and timescale objectives are satisfied.

RequirementCollection Resource

A collection of resources which constitute some statement of need.

Client

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

Server

An implementation of the OSLC Requirement Management specifications as a server. OSLC RM 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.

1.2 References

1.2.1 Normative references

[OSLCCore2]

S. Speicher; D. Johnson. [OSLC Core 2.0](http://open-services.net/bin/view/Main/OslcCoreSpecification). Finalized. URL: <http://open-services.net/bin/view/Main/OslcCoreSpecification>

[OSLCCore3]

Jim Amsden; S. Speicher. [OSLC Core 3.0](https://docs.oasis-open.org/oslc-core/oslc-requirements-management-spec). Project Specification. URL: [https://docs.oasis-open.org/oslc-core/oslc-](https://docs.oasis-open.org/oslc-core/oslc-requirements-management-spec)

[core/v3.0/oslc-core-v3.0-part1-overview.html](http://docs.oasis-open.org/oslc-core/v3.0/oslc-core-v3.0-part1-overview.html)

[OSLCCoreVocab]

Jim Amsden; S. Speicher. *OSLC Core 3.0 Vocabulary*. Project Specification. URL: <http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd03/part7-core-vocabulary/oslc-core-v3.0-csprd03-part7-core-vocabulary.html>

[OSLCResourcePreview]

Jim Amsden; S. Speicher. *OSLC Core 3.0 Resource Preview*. Project Specification. URL: <http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd03/part3-resource-preview/oslc-core-v3.0-csprd03-part3-resource-preview.html>

[OSLCShapes]

Arthur Ryman; Jim Amsden. *OSLC Resource Shape 3.0*. Project Specification. URL: <http://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/>

[RFC2119]

S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. March 1997. Best Current Practice. URL: <https://tools.ietf.org/html/rfc2119>

1.2.2 Informative references

[LDPPatch]

Linked Data Patch Format. Working Group Note. URL: <http://www.w3.org/TR/ldpatch/>

1.3 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].

2. Base Requirements

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

2.1 Base Requirements

This specification is based on [OSLCCore3]. OSLC RM 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. [cc-1]

An OSLC RM server **MUST** implement the domain vocabulary defined in [OSLC Requirements Management Version 2.1. Part 2: Vocabulary](#) [cc-2]

The following table summarizes the requirements from OSLC Core Specification as well as some additional requirements specific to the RM domain. Note that this specification further restricts some of the requirements for OSLC Core Specification. See the previous 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 servers MAY ignore unknown content and OSLC clients MUST preserve unknown content [cc-3]
Resource Operations	OSLC servers MUST support resource operations via standard HTTP operations [cc-4]
Resource Paging	OSLC servers MAY provide paging for resources but only when specifically requested by client [cc-5]
Partial Resource Representations	OSLC servers MUST support request for a subset of a resource's properties via the osc.properties URL parameter retrieval via HTTP GET and MAY support via HTTP PUT [cc-6]
Partial Update	OSLC servers MAY support partial update of resources using [LDPPatch]. [cc-7]
Discovery	OSLC servers MAY provide a Service Provider Catalog, MUST provide a Service Provider resource, and MAY provide other forms of discovery described in Core 3.0 Discovery. [cc-8]
Creation Factories	OSLC servers MUST provide at least one creation factory resource for requirements and MAY provide creation factory resources for requirement collections [cc-9]
Query Capabilities	OSLC servers MUST provide query capabilities to enable clients to query for resources [cc-10]
Query Syntax	OSLC query capabilities MUST support the OSLC Core Query Syntax [cc-11]
Delegated UI Dialogs	OSLC Services MUST offer delegated UI dialogs (for both creation and selection) specified via service provider resource [cc-12]
UI Preview	OSLC Services SHOULD offer UI previews for resources that may be referenced by other resources [cc-13]
HTTP Basic Authentication	OSLC Servers MAY support Basic Authentication and SHOULD only do so only over HTTPS [cc-14]
OAuth Authentication	OSLC Server MAY support OAuth and MAY indicate the required OAuth URLs via the service provider resource [cc-15]
Error Responses	OSLC Servers MAY provide error responses using Core defined error formats [cc-16]

Requirement	Meaning
RDF/XML Representations	OSLC servers MUST support RDF/XML representations for OSLC Defined Resources [cc-17]
XML Representations	OSLC servers MUST support XML representations that conform to the OSLC Core Guidelines for XML [cc-18]
JSON Representations	OSLC servers MAY support JSON representations; those which do MUST conform to the OSLC Core Guidelines for JSON [cc-19]
HTML Representations	OSLC servers MAY provide HTML representations for GET requests [cc-20]

2.2 Specification Versioning

This specification follows the specification version guidelines given in [OSLCCore3].

2.3 Namespaces

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

2.4 Resource Formats

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

For HTTP GET/PUT/POST requests on all OSLC RM and OSLC Core defined resource types,

- RM Servers **MUST** support RDF/XML representations with media-type `application/rdf+xml`. RM Clients **MUST** be prepared to deal with any valid RDF/XML document.
- RM Servers **MUST** support XML representations with media-type `application/xml`. The XML representations **MUST** follow the guidelines outlined in the [OSLC Core Representations Guidance](#) to maintain compatibility with [OSLCCore2].
- RM Servers **MAY** support JSON representations with media-type `application/json`. The JSON representations **MUST** follow the guidelines outlined in the [OSLC Core Representations Guidance](#) to maintain compatibility with [OSLCCore2].

[cc-21]

Additionally, for HTTP GET,

- RM Servers **SHOULD** provide an [X]HTML representation and a user interface (UI) preview as defined by [UI Preview Guidance](#)

[cc-22]

For HTTP GET response formats for Query requests,

- RM Servers **MUST** support RDF/XML representations with media-type `application/rdf+xml`.
- RM Servers **MUST** support XML representations with media-type `application/xml`.
- RM Servers **MAY** support JSON representations with media-type `application/json`.

[cc-23]

OSLC Servers **MAY** refuse to accept RDF/XML documents which do not have a top-level `rdf:RDF` document element. The OSLC Core describes an example, non-normative algorithm for generating RDF/XML representations of OSLC Defined Resources. [cc-24]

In addition to the resource formats defined above, Servers **MAY** support additional resource formats; the meaning and usage of these resource formats is not defined by this specification. [cc-25]

2.5 Authentication

[OSLCCore3] specifies the recommended OSLC authentication mechanisms. In addition to the OSLC Core authentication requirements, OSLC RM servers **SHOULD** support [OpenIDConnect]. [cc-26]

2.6 Error Responses

[OSLCCoreVocab] specifies the OSLC Core error responses. OSLC RM puts no additional constraints on error responses.

2.7 Pagination

OSLC RM servers **SHOULD** support pagination of query results and **MAY** support pagination of a single resource's properties as defined by [OSLCCore3]. [cc-27]

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. [cc-28]

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. [cc-29]

For compatibility with [OSLCCore2], a Server **MAY** also support partial update by identifying those properties to be modified using the `oslc.properties` URL parameter on a HTTP PUT request. [cc-30]

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. [cc-31]

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 RM servers **MAY** provide support for a partial update of the multi-valued properties as defined by draft specification [LDPPatch]. RM servers **MAY** also support partial updates through HTTP PUT where only the updated properties are included in the entity request body. [cc-32]

2.9 Labels for Relationships

This section is non-normative.

Requirement Management 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 [OSLCResourcePreview] to

Standards Track Work Product

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). Where the link requires a unique label that is not available from the target resource, 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 oslc_rm: <http://open-services.net/ns/rm#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

<http://example.com/requ/4321>
  a <http://open-services.net/ns/rm#Requirement> ;
  oslc_rm:elaboratedBy <http://anotherexample.com/requ/123> .

<http://njh.me/#link1>
  a rdf:Statement ;
  rdf:subject <http://example.com/requ/4321> ;
  rdf:predicate oslc_rm:elaboratedBy ;
  rdf:object <http://anotherexample.com/requ/123> ;
  dcterms:title "Requirement 123: The system shall be robust" .
```

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_rm": "http://open-services.net/ns/rm#"
  },
  "@id": "http://example.com/requ/4321",
  "@type": "oslc_rm:Requirement",
  "oslc_rm:elaboratedBy": {
    "@id": "http://anotherexample.com/requ/123",
    "dcterms:title": "Requirement 123: The system shall be robust"
  }
}
```

3. Vocabulary Terms and Constraints

[OSLC Requirements Management Version 2.1. Part 2: Vocabulary](#) defines the vocabulary terms and constraints for OSLC Requirements Management resources. These terms and constraints are specified according to [[OSLCCoreVocab](#)].

4. RM Server Capabilities

4.1 Server Resources

RM Servers **MUST** provide one or more `oslc:ServiceProvider` resources. Discovery of OSLC Service Provider Resources **MAY** be via one or more OSLC Service Provider Catalog Resources, or may be discovered by some other and/or additional Provider-specific means beyond the scope of this specification. The `oslc:Service` resources referenced by this `oslc:ServiceProvider` **MUST** have an `oslc:domain` of `http://open-services.net/ns/rm#`. [cc-33]

RM servers **MAY** provide other forms of discovery described in Core 3.0 Discovery. [cc-34]

RM Servers **MAY** provide one more more `oslc:ServiceProviderCatalog` resources. Any such catalog resources **MUST** include at least one `oslc:domain` of `http://open-services.net/ns/rm#`. Discovery of top-level OSLC Service Provider Catalog Resources is beyond the scope of this specification. [cc-35]

Service providers **MUST** give an `oslc:serviceProvider` property on all OSLC Defined Resources. This property **MUST** refer to an appropriate `oslc:ServiceProvider` resource. [cc-36]

4.2 Creation Factories

RM Servers supporting resource creation **MUST** do so through `oslc:CreationFactory` resources, as defined by [OSLCCore3]. Any such factory resources **MUST** be discoverable through `oslc:Service` resources. Servers **SHOULD** provide `oslc:ResourceShape` resources on `oslc:CreationFactory` resources as defined by [OSLCShapes]. [cc-37]

4.3 Query Capabilities

RM Servers **MUST** support query capabilities, as defined by [OSLCCore3]. Servers **SHOULD** provide `oslc:ResourceShape` on `oslc:QueryCapability` resources as defined by [OSLCShapes]. [cc-38]

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

- `oslc.where`
- `oslc.select`
- `oslc.properties`
- `oslc.prefix`

[cc-39]

Where `oslc:ResourceShape` is not supported by the Query Capability, Servers **SHOULD** use the following guidance to represent query results: [cc-40]

- For RDF/XML and XML, use `rdf:Description` and `rdfs:member` as defined by [Core Specification Appendix B: Representations and Examples - RDF/XML Examples](#).
- For JSON the query results are contained within `oslc:results` array. See [Core Specification Appendix B: Representations and Examples - Guidelines for JSON](#).

The stability of query results is **OPTIONAL** (see [Core Specification Version 2.0 - Stable Paging](#)).

4.4 Delegated UIs

RM Servers **MUST** support the selection and creation of resources by delegated web-based user interface dialogs [Delegated Dialogs](#) as defined by [OSLCCore3]. [cc-41]

RM Servers **MAY** support the pre-filling of creation dialogs based on the definition at [Delegated Dialogs](#). [cc-42]

4.5 Usage Identifiers

RM Servers **MAY** identify the usage of various services with additional property values for the [OSLC Core Discovery](#) defined `oslc:usage` property on `oslc:Dialog`, `CreationFactory` and `QueryCapability`. The `oslc:usage` property value of <http://open-services.net/ns/core#default> **SHOULD** be used to designate the default or primary service to be used by consumers when multiple entries are found. [cc-43]

There are no additional usage identifiers defined by this specification. RM Servers **MAY** provide their own usage URIs. Such usage URIs **MUST** be in a non-OSLC namespace. [cc-44]

5. Conformance

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

Clause Number	Requirement
cc-1	This specification is based on [OSLCCore3] . OSLC RM 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.
cc-2	An OSLC RM server MUST implement the domain vocabulary defined in OSLC Requirements Management Version 2.1. Part 2: Vocabulary
cc-3	OSLC servers MAY ignore unknown content and OSLC clients MUST preserve unknown content
cc-4	OSLC servers MUST support resource operations via standard HTTP operations
cc-5	OSLC servers MAY provide paging for resources but only when specifically requested by client
cc-6	OSLC servers MUST support request for a subset of a resource's properties via the <code>oslc.properties</code> URL parameter retrieval via HTTP GET and MAY support via HTTP PUT
cc-7	OSLC servers MAY support partial update of resources using [LDPPatch] .
cc-8	OSLC servers MAY provide a Service Provider Catalog, MUST provide a Service Provider resource, and MAY provide other forms of discovery described in Core 3.0 Discovery.
cc-9	OSLC servers MUST provide at least one creation factory resource for requirements and MAY provide creation factory resources for requirement collections
cc-10	OSLC servers MUST provide query capabilities to enable clients to query for resources
cc-11	OSLC query capabilities MUST support the OSLC Core Query Syntax
cc-12	OSLC Services MUST offer delegated UI dialogs (for both creation and selection) specified via service provider resource
cc-13	OSLC Services SHOULD offer UI previews for resources that may be referenced by other resources
cc-14	OSLC Servers MAY support Basic Authentication and SHOULD only do so only over HTTPS
cc-15	OSLC Server MAY support OAuth and MAY indicate the required OAuth URLs via the service provider resource
cc-16	OSLC Servers MAY provide error responses using Core defined error formats
cc-17	OSLC servers MUST support RDF/XML representations for OSLC Defined Resources
cc-18	OSLC servers MUST support XML representations that conform to the OSLC Core Guidelines for XML
cc-19	OSLC servers MAY support JSON representations; those which do MUST conform to the OSLC Core Guidelines for JSON
cc-20	OSLC servers MAY provide HTML representations for GET requests
cc-21	<p>For HTTP GET/PUT/POST requests on all OSLC RM and OSLC Core defined resource types,</p> <ul style="list-style-type: none"> • RM Servers MUST support RDF/XML representations with media-type <code>application/rdf+xml</code>. RM Clients MUST be prepared to deal with any valid RDF/XML document. • RM Servers MUST support XML representations with media-type <code>application/xml</code>. The XML representations MUST follow the guidelines outlined in the OSLC Core Representations Guidance to maintain compatibility with [OSLCCore2]. • RM Servers MAY support JSON representations with media-type <code>application/json</code>. The JSON representations MUST follow the guidelines outlined in the OSLC Core Representations Guidance to maintain compatibility with [OSLCCore2].

Standards Track Work Product

Clause Number	Requirement
cc-22	<ul style="list-style-type: none"> RM Servers SHOULD provide an [X]HTML representation and a user interface (UI) preview as defined by UI Preview Guidance
cc-23	<p>For HTTP GET response formats for Query requests,</p> <ul style="list-style-type: none"> RM Servers MUST support RDF/XML representations with media-type <code>application/rdf+xml</code>. RM Servers MUST support XML representations with media-type <code>application/xml</code>. RM Servers MAY support JSON representations with media-type <code>application/json</code>.
cc-24	OSLC Servers MAY refuse to accept RDF/XML documents which do not have a top-level <code>rdf:RDF</code> document element. The OSLC Core describes an example, non-normative algorithm for generating RDF/XML representations of OSLC Defined Resources.
cc-25	In addition to the resource formats defined above, Servers MAY support additional resource formats; the meaning and usage of these resource formats is not defined by this specification.
cc-26	[OSLCCore3] specifies the recommended OSLC authentication mechanisms. In addition to the OSLC Core authentication requirements, OSLC RM servers SHOULD support [OpenIDConnect].
cc-27	OSLC RM servers SHOULD support pagination of query results and MAY support pagination of a single resource's properties as defined by [OSLCCore3].
cc-28	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.
cc-29	A client MAY request that a subset of a resource's properties be updated by using the [LDPPatch] PATCH method.
cc-30	For compatibility with [OSLCCore2], a Server 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.
cc-31	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.
cc-32	For multi-valued properties that contain a large number of values, it may be difficult and inefficient to add or remove property values. OSLC RM servers MAY provide support for a partial update of the multi-valued properties as defined by draft specification [LDPPatch]. RM servers MAY also support partial updates through HTTP PUT where only the updated properties are included in the entity request body.
cc-33	RM Servers MUST provide one or more <code>oslc:ServiceProvider</code> resources. Discovery of OSLC Service Provider Resources MAY be via one or more OSLC Service Provider Catalog Resources, or may be discovered by some other and/or additional Provider-specific means beyond the scope of this specification. The <code>oslc:Service</code> resources referenced by this <code>oslc:ServiceProvider</code> MUST have an <code>oslc:domain</code> of <code>http://open-services.net/ns/rm#</code> .
cc-34	RM servers MAY provide other forms of discovery described in Core 3.0 Discovery.
cc-35	RM Servers MAY provide one or more <code>oslc:ServiceProviderCatalog</code> resources. Any such catalog resources MUST include at least one <code>oslc:domain</code> of <code>http://open-services.net/ns/rm#</code> . Discovery of top-level OSLC Service Provider Catalog Resources is beyond the scope of this specification.
cc-36	Service providers MUST give an <code>oslc:serviceProvider</code> property on all OSLC Defined Resources. This property MUST refer to an appropriate <code>oslc:ServiceProvider</code> resource.
cc-37	RM Servers supporting resource creation MUST do so through <code>oslc:CreationFactory</code> resources, as defined by [OSLCCore3]. Any such factory resources MUST be discoverable through <code>oslc:Service</code> resources. Servers SHOULD provide <code>oslc:ResourceShape</code> resources on <code>oslc:CreationFactory</code> resources as defined by [OSLCShapes].

Standards Track Work Product

Clause Number	Requirement
cc-38	RM Servers MUST support query capabilities, as defined by [OSLCCore3]. Servers SHOULD provide <code>oslc:ResourceShape</code> on <code>oslc:QueryCapability</code> resources as defined by [OSLCShapes].
cc-39	<p>The Query Capability, if supported, MUST support these parameters:</p> <ul style="list-style-type: none"> • <code>oslc.where</code> • <code>oslc.select</code> • <code>oslc.properties</code> • <code>oslc.prefix</code>
cc-40	Where <code>oslc:ResourceShape</code> is not supported by the Query Capability, Servers SHOULD use the following guidance to represent query results:
cc-41	RM Servers MUST support the selection and creation of resources by delegated web-based user interface dialogs Delegated Dialogs as defined by [OSLCCore3].
cc-42	RM Servers MAY support the pre-filling of creation dialogs based on the definition at Delegated Dialogs .
cc-43	RM Servers MAY identify the usage of various services with additional property values for the OSLC Core Discovery defined <code>oslc:usage</code> property on <code>oslc:Dialog</code> , <code>CreationFactory</code> and <code>QueryCapability</code> . The <code>oslc:usage</code> property value of <code>http://open-services.net/ns/core#default</code> SHOULD be used to designate the default or primary service to be used by consumers when multiple entries are found.
cc-44	There are no additional usage identifiers defined by this specification. RM Servers MAY provide their own usage URIs. Such usage URIs MUST be in a non-OSLC namespace.

Appendix A. Version Compatibility

A.1 Version Compatibility with 2.0 Specifications

This section is non-normative.

The specification is updated to be based on the [OSLCCore3] Specification. The changes are all upward compatible additions and therefore do not introduce incompatibilities with version 2.0.

A.2 Version Compatibility with 1.0 Specifications

This section is non-normative.

The goal is to provide a smooth transition to 2.0 for both Clients and Servers. This section will clarify the usage of 1.0 media types so that Servers can support both 1.0 and 2.0 Clients when HTTP requests are made for a resource with the same URI.

Network addressable resource URIs used for 1.0 resources for these types: Requirement, RequirementCollection, ServiceDescriptor and ServiceProviderCatalog, should not have to change. Clients who support both 1.0 and 2.0, should only preserve these resource URIs. When a Server starts to serve 2.0 resource formats, for instance the ServiceProvider resource, it is recommended to update its locally stored or cached information about the contents of the ServiceProvider resource as the URIs to various capabilities may have changed (query, delegated Uls, factories, etc.).

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 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

Additional Participants:

Andy Berner, IBM
Scott Bosworth, IBM
Jim Conallen, IBM
George De Candio, IBM
Jeremy Dick, Integrate
Brenda Ellis, Northrop Grumman
Rainer Ersch, Siemens
Ian Green, IBM
Dave Johnson, IBM
Andreas Keis, EADS
Nicholas Kruk, IBM
Chris McGraw, IBM
Paul McMahan, IBM
David Ruiz, Ravenflow
Matthew Stone, Stoneworks
Dominic Tulley, IBM
Simon Wills, Integrate

Appendix C. Change History

This section is non-normative.

Revision	Date	Editor	Changes Made
01	2018-05-22	Jad El-khoury	Initial Committee Specification Draft migrated from open-services.net