



OSLC Change Management Version 3.0. Part 1: Specification

Project Specification Draft 03 15 August 2019

This stage:

<https://docs.oasis-open-projects.org/oslc-op/cm/v3.0/psd03/change-mgt-spec.html> (Authoritative)
<https://docs.oasis-open-projects.org/oslc-op/cm/v3.0/psd03/change-mgt-spec.pdf>

Previous stage:

<http://docs.oasis-open.org/oslc-domains/cm/v3.0/cs02/part1-change-mgt/cm-v3.0-cs02-part1-change-mgt.html> (Authoritative)
<http://docs.oasis-open.org/oslc-domains/cm/v3.0/cs02/part1-change-mgt/cm-v3.0-cs02-part1-change-mgt.pdf>

Latest stage:

<https://docs.oasis-open-projects.org/oslc-op/cm/v3.0/change-mgt-spec.html> (Authoritative)
<https://docs.oasis-open-projects.org/oslc-op/cm/v3.0/change-mgt-spec.pdf>

Latest version:

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

Latest editor's draft:

<https://open-services.net/spec/cm/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](#)

Editor:

Jim Amsden (jamsden@us.ibm.com), [IBM](#)

Additional components:

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

- OSLC Change Management Version 3.0. Part 1: Specification (this document). <https://docs.oasis-open-projects.org/oslc-op/cm/v3.0/psd03/change-mgt-spec.html>
- OSLC Change Management Version 3.0. Part 2: Vocabulary. <https://docs.oasis-open-projects.org/oslc-op/cm/v3.0/psd03/change-mgt-vocab.html>

Related work:

This specification is related to:

- *Open Services for Lifecycle Collaboration Change Management Specification Version 2.0.* <http://open-services.net/bin/view/Main/CmSpecificationV2>

RDF Namespaces:

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

Abstract:

This specification defines the OSLC Change Management domain, a RESTful web services interface for the management of product change requests, activities, tasks and relationships between those and related resources such as requirements, test cases, 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-CM-3.0-Part1]

OSLC Change Management Version 3.0. Part 1: Specification. Edited by Jim Amsden. 15 August 2019. OASIS Project Specification Draft 03. <https://docs.oasis-open-projects.org/oslc-op/cm/v3.0/psd03/change-mgt-spec.html>. Latest stage: <https://docs.oasis-open-projects.org/oslc-op/cm/v3.0/change-mgt-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 Compliance](#)
 - 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 [Status, State and State Predicates](#)
 - 2.10 [Labels for Relationships](#)
- 3. [Vocabulary Terms and Constraints](#)
- 4. [CM Server Capabilities](#)
 - 4.1 [Server Resources](#)
 - 4.2 [Creation Factories](#)
 - 4.3 [Query Capabilities](#)
 - 4.4 [Delegated UIs](#)
 - 4.5 [Usage Identifiers](#)
- 5. [Conformance](#)
- Appendix A. [Version Compatibility with 2.0 Specifications](#)
 - A.1 [Deprecated terms](#)
 - A.2 [Changes from 2.0](#)
- Appendix B. [Acknowledgements](#)

1. Introduction

This section is non-normative.

This specification defines the OSLC Change management domain as a RESTful web services interface for the management of product change requests, activities, tasks and relationships between those and related resources such as project, category, release and plan. 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.

The intent of this specification is to define the capabilities needed to support integration scenarios defined by the OASIS OSLC Open Project and not to provide a comprehensive interface to Change Management. The resource formats and operations may not match exactly the native models supported by change management servers but are intended to be compatible with them. The approach to supporting these scenarios is to delegate operations, as driven by server contributed user interfaces, as much as possible and not require a server to expose its complete data model and application logic.

1.1 Terminology

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

Change Request Resource

A request for change to an application or product. Typically a product request for enhancement, a report for a resolution of a product defect or simply a bug report.

Client

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

Server

An implementation of the OSLC Change Management specifications as a server. OSLC CM 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-core/v3.0/oslc-core-v3.0-part1-overview.html). Project Specification. URL: <https://docs.oasis-open.org/oslc-core/oslc-core/v3.0/oslc-core-v3.0-part1-overview.html>

[OSLCCoreVocab]

Jim Amsden; S. Padgett; S. Speicher. [OSLC Core Vocabulary](http://open-services.net/specifications/core/core-vocab.html). Project Specification. URL: <http://open-services.net/specifications/core/core-vocab.html>

[OSLCResourcePreview]

Jim Amsden; S. Speicher. [OSLC Core 3.0 Resource Preview](http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd03/part3-resource-preview/oslc-core-v3.0-csprd03-part3-resource-preview.html). 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/>

[OSLCActions]

Martin Pain; Steve Speicher. *Open Services for Lifecycle Collaboration Actions Specification Version 2.0*. Finalization. URL: <http://open-services.net/wiki/core/Actions-2.0/>

[OSLCCM20]

S. Speicher. *Open Services for Lifecycle Collaboration Change Management Specification Version 2.0*. Finalized. URL: <http://open-services.net/bin/view/Main/CmSpecificationV2>

[OSLCQM]

Paul McMahan. *Open Services for Lifecycle Collaboration Quality Management Specification Version 2.0*. Final. URL: <http://open-services.net/bin/view/Main/QmSpecificationV2>

[OSLCRM]

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

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 Compliance

This specification is based on [\[OSLCCore3\]](#). OSLC CM 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. [\[cm-1\]](#)

An OSLC CM server **MUST** implement the domain vocabulary defined in [OSLC Change Management Version 3.0. Part 2: Vocabulary](#) [\[cm-2\]](#)

The following table summarizes the requirements from OSLC Core Specification as well as some additional requirements specific to the CM 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 [cm-3]
Resource Operations	OSLC service MUST support resource operations via standard HTTP operations [cm-4]
Resource Paging	OSLC servers MAY provide paging for resources but only when specifically requested by client [cm-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 [cm-6]
Partial Update	OSLC servers MAY support partial update of resources using [LDPPatch] , or via HTTP PUT. [cm-7]
Discovery	OSLC servers SHOULD provide a ServiceProvider resource for Core v2 compatibility, MAY provide a ServiceProviderCatalog, and MAY provide other forms of discovery described in Core 3.0 Discovery. [cm-8]
Creation Factories	OSLC servers MUST provide LDPC creation factories to enable resource creation of Change Management resources via HTTP POST [cm-9]
Query Capabilities	OSLC servers SHOULD provide query capabilities to enable clients to query for resources [cm-10]
Query Syntax	OSLC query capabilities SHOULD support the OSLC Core Query Syntax and MAY use other query syntax [cm-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 [cm-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 [cm-13]
Authentication	OSLC Services SHOULD follow the recommendations for Authentication specified in [OSLCCore3] [cm-14]

Requirement	Meaning
Error Responses	OSLC Services SHOULD provide error responses using OSLC Core 3.0 defined error formats [cm-15]
Turtle Representations	OSLC servers MUST provide a Turtle representation for HTTP GET requests and SHOULD support Turtle representations on POST and PUT requests. [cm-16]
RDF/XML Representations	OSLC servers SHOULD provide an RDF/XML representation for HTTP GET requests and SHOULD support RDF/XML representations on POST and PUT requests for compatibility with Change Management 2.0. [cm-17]
XML Representations	OSLC servers SHOULD provide a XML representation for HTTP GET, POST and PUT requests that conform to the Core 2.0 Guidelines for XML. [cm-18]
JSON Representations	OSLC servers MUST provide JSON-LD representations for HTTP GET, POST and PUT requests that conform to the Core Guidelines for JSON-LD [cm-19]
HTML Representations	OSLC servers SHOULD provide HTML representations for HTTP GET requests [cm-20]

2.2 Specification Versioning

This specification follows the specification version guidelines given in [\[OSLCCore3\]](#).

2.3 Namespaces

In addition to the namespace URLs and namespace prefixes **oslc**, **rdf**, **dcterms** and **foaf** defined in the [\[OSLCCore3\]](#), OSLC CM defines the namespace URI of <http://open-services.net/ns/cm#> with a namespace prefix of **oslc_cm**.

This specification also uses these namespace prefix definitions:

- **oslc_rm** : <http://open-services.net/ns/rm#> [\[OSLCRM\]](#)
- **oslc_qm** : <http://open-services.net/ns/qm#> [\[OSLCQM\]](#)

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 CM and OSLC Core defined resource types,

- CM 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 [\[OSLCCore2\]](#). [\[cm-21\]](#)
- CM clients requesting RDF/XML **SHOULD** be prepared for any valid RDF/XML document. CM clients requesting XML **SHOULD** be prepared for representations that follow the guidelines outlined in the [OSLC Core Representations Guidance](#). [\[cm-22\]](#)
- CM servers **SHOULD** support an [X]HTML representation and a user interface (UI) preview as defined by [UI Preview Guidance](#) [\[cm-23\]](#)

For HTTP PUT/POST request formats for resource type of ChangeRequest:

- CM servers **MUST** accept Turtle and JSON-LD representations and **SHOULD** accept RDF/XML and XML representations. CM servers accepting RDF/XML **SHOULD** be prepared for any valid RDF/XML document. For XML CM servers **SHOULD** be prepared for representations that follow the guidelines outlined in the [OSLC Core Representations Guidance](#). [\[cm-24\]](#)

For HTTP GET response formats for Query requests,

CM servers **MUST** provide Turtle and JSON-LD, **SHOULD** provide RDF/XML, XML, and **MAY** provide Atom Syndication Format XML representations. [\[cm-25\]](#)

When CM clients request:

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

[cm-26]

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 CM 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 CM servers **SHOULD** support [OpenIDConnect]. [cm-27]

2.6 Error Responses

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

2.7 Pagination

OSLC CM servers **SHOULD** support pagination of query results and **MAY** support pagination of a single resource's properties as defined by [OSLCCore3]. [cm-28]

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. [cm-29]

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. [cm-30]

For compatibility with [OSLCCore2], CM servers class='conformance' also support partial update by identifying those properties to be modified using the `oslc.properties` URL parameter on a HTTP PUT request. [cm-31]

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. [cm-32]

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 CM servers **MAY** provide support for a partial update of the multi-valued properties as defined by draft specification [LDPPatch]. CM servers **MAY** also support partial updates through HTTP PUT where only the updated properties are included in the entity request body. [cm-33]

2.9 Status, State and State Predicates

Probably the most important property of a Change Request is the status property. "Status" specifies the location of a Change Request in a workflow. The `oslc_cm:status` property is a typically read-only string that servers can set to describe the status of a ChangeRequest. In queries the `oslc_cm:status` property can be used to filter change request (e.g. all change requests that are "fixed") and may be used to perform state transitions (not part of this specification) on a change request, e.g. closing a change request as "fixed".

The problem is that different CM servers may use different properties (or even a set of properties) and different values to represent the change request's state. Even providing access to meta data does not help because knowing all possible state values does not reveal the semantics of a state.

[OSLCCM20] introduced State Predicates to define single-value often read-only Boolean properties on a Change Request resource that allow servers to translate their specific status string values into a standard boolean state predicate value. For example, a CM server could set `oslc_cm:status` to "BeingInvestigated", and map this status to the following state predicate values:

- `oslc_cm:closed=false`
- `oslc_cm:inProgress=true`
- `oslc_cm:fixed=false`
- `oslc_cm:approved=false`
- `oslc_cm:reviewed=false`
- `oslc_cm:verified=false`

An attempt to update read-only predicates **SHOULD** be answered with a 409 Conflict HTTP status code. Their presence in a resource representation used for an update via PUT **MUST NOT** prevent the resource from being updated. Predicates **MUST** be queryable. The Change Request resource definition sections defines the complete set of predicates. [cm-34]

OSLC CM 3.0 provides a similar means of defining a set of standard states that is also extensible. The `oslc_cm:state` property can have a value of type `oslc_cm:State` which represents an enumeration of individuals representing a standard set of ChangeRequest states. Servers **MAY** define additional individuals to extend the states of different ChangeRequest types and lifecycles. [cm-35]

2.10 Labels for Relationships

This section is non-normative.

Change 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 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/cm#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

<http://example.com/bugs/4321>
  a <http://open-services.net/ns/cm#ChangeRequest> ;
  ns0:relatedChangeRequest <http://anotherexample.com/defects/123> .

<http://njh.me/#link1>
  a rdf:Statement ;
  rdf:subject <http://example.com/bugs/4321> ;
  rdf:predicate ns0:relatedChangeRequest ;
  rdf:object <http://anotherexample.com/defects/123> ;
  dcterms:title "Defect 123: Problems during install" .
```

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_cm": "http://open-services.net/ns/cm#"
  },
  "@id": "http://example.com/bugs/4321",
  "@type": "oslc_cm:ChangeRequest",
  "oslc_cm:relatedChangeRequest": {
    "@id": "http://anotherexample.com/defects/123",
    "dcterms:title": "Defect 123: Problems during install"
  }
}
```

3. Vocabulary Terms and Constraints

[OSLC Change Management Version 3.0. Part 2: Vocabulary](#) Defines the vocabulary terms and constraints for OSLC Change Management resources. These terms and constraints are specified according to [[OSLCCoreVocab](#)].

4. CM Server Capabilities

4.1 Server Resources

OSLC CM servers **MUST** support OSLC Discovery capabilities defined by [OSLCCore3]. [cm-36]

OSLC CM servers **MAY** provide a ServiceProvider Resource that can be retrieved at a implementation dependent URI. [cm-37]

OSLC CM servers **MAY** provide a ServiceProviderCatalog Resource that can be retrieved at a implementation dependent URI. [cm-38]

OSLC CM servers **MAY** provide a `oslc:serviceProvider` property for their defined resources that will be the URI to a ServiceProvider Resource. [cm-39]

OSLC CM servers **MUST** supply a value of `http://open-services.net/ns/cm#` for the property `oslc:domain` on either `oslc:Service` or `oslc:ServiceProviderCatalog` resources. [cm-40]

OSLC CM servers **MAY** allow ChangeRequest state change through [OSLCActions]. [cm-41]

4.2 Creation Factories

OSLC CM servers **MUST** support [Creation Factories](#) and list them in the Service Provider Resource as defined by OSLC Core.

OSLC CM servers **SHOULD** support Resource Shapes for Creation Factories as defined in [OSLCShapes] [cm-42]

4.3 Query Capabilities

OSLC CM servers **SHOULD** support the [Query Capabilities](#) as defined by [OSLCCore3]. OSLC CM servers **SHOULD** support Resource Shapes for Query Capability as defined in [OSLCShapes] [cm-43]

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

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

[cm-44]

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](#)

[cm-45]

4.4 Delegated UIs

OSLC CM servers **MUST** support the selection and creation of resources by delegated web-based user interface dialogs [Delegated UIs](#) as defined by [OSLCCore3]. [cm-46]

OSLC CM servers **MAY** support the pre-filling of creation dialogs based on the definition at [Delegated UIs](#). [cm-47]

4.5 Usage Identifiers

An OSLC CM server can identify or distinguish the usage of various services with additional property values for the [OSLC Core](#)

defined `oslc:usage` property on `oslc:Dialog`, `CreationFactory` and `QueryCapability`. The `oslc:usage` property value of `http://open-services.net/ns/core#default` will be used to designate the default or primary service to be used by clients when multiple entries are found.

The additional Change Management property values for `oslc:usage` are:

- `http://open-services.net/ns/cm#defect` - primarily used by QM tools to report defects in testing.
- `http://open-services.net/ns/cm#planItem` - used by QM and PPM tools for associating change requests into plans (project, release, sprint, etc).
- `http://open-services.net/ns/cm#task` - used by QM and PPM tools for associating change requests into executable and track-able items.
- `http://open-services.net/ns/cm#requirementsChangeRequest` - used by RM tools for associating a change request for usage in tracking changes to a Requirements resource

5. Conformance

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

Clause Number	Requirement
cm-1	This specification is based on [OSLCCore3] . OSLC CM 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.
cm-2	An OSLC CM server MUST implement the domain vocabulary defined in OSLC Change Management Version 3.0. Part 2: Vocabulary
cm-3	OSLC servers MAY ignore unknown content and OSLC clients MUST preserve unknown content
cm-4	OSLC service MUST support resource operations via standard HTTP operations
cm-5	OSLC servers MAY provide paging for resources but only when specifically requested by client
cm-6	OSLC servers MUST support request for a subset of a resource's properties via the osc.properties URL parameter retrieval via HTTP GET
cm-7	OSLC servers MAY support partial update of resources using [LDPPatch] , or via HTTP PUT.
cm-8	OSLC servers SHOULD provide a ServiceProvider resource for Core v2 compatibility, MAY provide a ServiceProviderCatalog, and MAY provide other forms of discovery described in Core 3.0 Discovery.
cm-9	OSLC servers MUST provide LDPC creation factories to enable resource creation of Change Management resources via HTTP POST
cm-10	OSLC servers SHOULD provide query capabilities to enable clients to query for resources
cm-11	OSLC query capabilities SHOULD support the OSLC Core Query Syntax and MAY use other query syntax
cm-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
cm-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
cm-14	OSLC Services SHOULD follow the recommendations for Authentication specified in [OSLCCore3]
cm-15	OSLC Services SHOULD provide error responses using OSLC Core 3.0 defined error formats
cm-16	OSLC servers MUST provide a Turtle representation for HTTP GET requests and SHOULD support Turtle representations on POST and PUT requests.
cm-17	OSLC servers SHOULD provide an RDF/XML representation for HTTP GET requests and SHOULD support RDF/XML representations on POST and PUT requests for compatibility with Change Management 2.0.
cm-18	OSLC servers SHOULD provide a XML representation for HTTP GET, POST and PUT requests that conform to the Core 2.0 Guidelines for XML.
cm-19	OSLC servers MUST provide JSON-LD representations for HTTP GET, POST and PUT requests that conform to the Core Guidelines for JSON-LD
cm-20	OSLC servers SHOULD provide HTML representations for HTTP GET requests
cm-21	CM 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] .
cm-22	CM clients requesting RDF/XML SHOULD be prepared for any valid RDF/XML document. CM clients requesting XML SHOULD be prepared for representations that follow the guidelines outlined in the OSLC Core Representations Guidance .
cm-23	CM servers SHOULD support an [X]HTML representation and a user interface (UI) preview as defined by UI Preview Guidance

Clause Number	Requirement
cm-24	CM servers MUST accept Turtle and JSON-LD representations and SHOULD accept RDF/XML and XML representations. CM servers accepting RDF/XML SHOULD be prepared for any valid RDF/XML document. For XML CM servers SHOULD be prepared for representations that follow the guidelines outlined in the OSLC Core Representations Guidance .
cm-25	CM servers MUST provide Turtle and JSON-LD, SHOULD provide RDF/XML, XML, and MAY provide Atom Syndication Format XML representations.
cm-26	When CM clients request: <ul style="list-style-type: none"> • <code>text/turtle</code> CM servers MUST respond with Turtle representation. • <code>application/ld+json</code> CM servers MUST respond with JSON-LD representation. • <code>application/rdf+xml</code> CM servers SHOULD respond with RDF/XML representation without restrictions. • <code>application/xml</code> CM servers SHOULD respond with OSLC-defined abbreviated XML representation as defined in the OSLC Core Representations Guidance • <code>application/atom+xml</code> CM servers SHOULD respond with Atom Syndication Format XML representation as defined in the OSLC Core Representations Guidance • The Atom Syndication Format XML representation SHOULD use RDF/XML representation without restrictions for the atom:content entries representing the resource representations.
cm-27	[OSLCCore3] specifies the recommended OSLC authentication mechanisms. In addition to the OSLC Core authentication requirements, OSLC CM servers SHOULD support [OpenIDConnect].
cm-28	OSLC CM servers SHOULD support pagination of query results and MAY support pagination of a single resource's properties as defined by [OSLCCore3].
cm-29	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.
cm-30	A client MAY request that a subset of a resource's properties be updated by using the [LDPPatch] PATCH method.
cm-31	For compatibility with [OSLCCore2], CM servers class='conformance' also support partial update by identifying those properties to be modified using the <code>oslc.properties</code> URL parameter on a HTTP PUT request.
cm-32	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.
cm-33	For multi-valued properties that contain a large number of values, it may be difficult and inefficient to add or remove property values. OSLC CM servers MAY provide support for a partial update of the multi-valued properties as defined by draft specification [LDPPatch]. CM servers MAY also support partial updates through HTTP PUT where only the updated properties are included in the entity request body.
cm-34	An attempt to update read-only predicates SHOULD be answered with a 409 Conflict HTTP status code. Their presence in a resource representation used for an update via PUT MUST NOT prevent the resource from being updated. Predicates MUST be queryable. The Change Request resource definition sections defines the complete set of predicates.
cm-35	OSLC CM 3.0 provides a similar means of defining a set of standard states that is also extensible. The <code>oslc_cm:state</code> property can have a value of type <code>oslc_cm:State</code> which represents an enumeration of individuals representing a standard set of ChangeRequest states. Servers MAY define additional individuals to extend the states of different ChangeRequest types and lifecycles.
cm-36	OSLC CM servers MUST support OSLC Discovery capabilities defined by [OSLCCore3].
cm-37	OSLC CM servers MAY provide a ServiceProvider Resource that can be retrieved at a implementation dependent URI.

Clause Number	Requirement
cm-38	OSLC CM servers MAY provide a ServiceProviderCatalog Resource that can be retrieved at a implementation dependent URI.
cm-39	OSLC CM servers MAY provide a <code>oslc:serviceProvider</code> property for their defined resources that will be the URI to a ServiceProvider Resource.
cm-40	OSLC CM servers MUST supply a value of <code>http://open-services.net/ns/cm#</code> for the property <code>oslc:domain</code> on either <code>oslc:Service</code> or <code>oslc:ServiceProviderCatalog</code> resources.
cm-41	OSLC CM servers MAY allow ChangeRequest state change through [OSLCActions] .
cm-42	OSLC CM servers MUST support Creation Factories and list them in the Service Provider Resource as defined by OSLC Core. OSLC CM servers SHOULD support Resource Shapes for Creation Factories as defined in [OSLCShapes]
cm-43	OSLC CM servers SHOULD support the Query Capabilities as defined by [OSLCCore3] . OSLC CM servers SHOULD support Resource Shapes for Query Capability as defined in [OSLCShapes]
cm-44	<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>
cm-45	<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 <code>rdf:Description</code> and <code>rdfs:member</code> as defined in OSLC Core RDF/XML Examples. • For JSON, the query results are contained within <code>oslc:results</code> array. See OSLC Core Representation Guidance for JSON
cm-46	OSLC CM servers MUST support the selection and creation of resources by delegated web-based user interface dialogs Delegated UIs as defined by [OSLCCore3] .
cm-47	OSLC CM servers MAY support the pre-filling of creation dialogs based on the definition at Delegated UIs .

Appendix A. Version Compatibility with 2.0 Specifications

A.1 Deprecated terms

A number of terms introduced in early development of the OSLC Change Management domain were deprecated in the finalized [OSLCCM20] specification. These terms are summarized here in order to indicate they remain deprecated.

Prefixed Name	Occurs	Read-only	Value-type	Representation	Range	Description
<i>deprecated</i> dcterms:type	zero-or-more	unspecified	String	n/a	n/a	A short string representation for the type, example Defect .
Relationship properties: This grouping of properties are used to identify relationships between resources managed by other OSLC servers						
<i>deprecated</i> osc_cm:testedByTestCase	zero-or-many	False	Resource	Reference	any	Test case by which this change request is tested. It is likely that the target resource will be an oslc_qm:TestCase , but that is not necessarily the case.
<i>deprecated</i> osc_cm:affectsTestResult	zero-or-many	False	Resource	Reference	any	Associated QM resource that is affected by this Change Request. It is likely that the target resource will be an oslc_qm:TestResult , but that is not necessarily the case.
<i>deprecated</i> osc_cm:blocksTestExecutionRecord	zero-or-many	False	Resource	Reference	any	Associated QM resource that is blocked by this Change Request. It is likely that the target resource will be an oslc_cm:TestExecutionRecord , but that is not necessarily the case.
<i>deprecated</i> osc_cm:relatedTestExecutionRecord	zero-or-many	False	Resource	Reference	any	Related to a QM test execution resource. It is likely that the target resource will be an oslc_qm:TestExecutionRecord , but that is not necessarily the case.
<i>deprecated</i> osc_cm:relatedTestCase	zero-or-many	False	Resource	Reference	any	Related QM test case resource. It is likely that the target resource will be an oslc_qm:TestCase , but that is not necessarily the case.

Prefixed Name	Occurs	Read-only	Value-type	Representation	Range	Description
<i>deprecated</i> osc_cm:relatedTestPlan	zero-or-many	False	Resource	Reference	<i>any</i>	Related QM test plan resource. It is likely that the target resource will be an <i>oslc_qm:TestPlan</i> , but that is not necessarily the case.
<i>deprecated</i> osc_cm:relatedTestScript	zero-or-many	False	Resource	Reference	<i>any</i>	Related QM test script resource. It is likely that the target resource will be an <i>oslc_qm:TestScript</i> , but that is not necessarily the case.

A.2 Changes from 2.0

The following lists the significant vocabulary changes that were introduced in this specification. The changes are all upward compatible additions and therefore do not introduce incompatibilities with version 2.0.

- Added *oslc_cm:priority* and *oslc_cm:severity*
- Added new change management types

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:

Gray Bachelor, IBM
Geoff Clemm, IBM
Peter Hack, IBM
Sam Padget, IBM
Martin Pain, IBM
Brian Steele, IBM
Jad El-Khoury, KTH
Martin Sarabura, PTC
Jory Burson, OASIS
Chet Ensign, OASIS
Carol Geyer, OASIS