Open Services for Lifecycle Collaboration
Change Management Specification Version 2.0

Status: 2.0 Specification - November 19, 2010

This Version
- [http://open-services.net/bin/view/Main/CmSpecificationV2](http://open-services.net/bin/view/Main/CmSpecificationV2)

Latest Version
- [http://open-services.net/bin/view/Main/CmSpecificationV2](http://open-services.net/bin/view/Main/CmSpecificationV2)

Previous Version
- [http://open-services.net/bin/view/Main/CmSpecificationV1](http://open-services.net/bin/view/Main/CmSpecificationV1)

Authors
- SteveSpeicher

Contributors
- See [Contributors section](http://open-services.net/bin/view/Main/CmSpecificationV2) below

Table of Contents

- Introduction
  - Terminology
- Base Requirements
  - Compliance
  - Specification Versioning
  - Namespaces
  - Resource Formats
  - Authentication
  - Error Responses
  - Pagination
  - Requesting and Updating Properties
  - State Predicates
  - Labels for Relationships
- CM Resource Definitions
  - Resource ChangeRequest
- CM Service Provider Capabilities
  - Service Provider Resources
  - Creation Factories
  - Query Capabilities
  - Delegated UIs
  - Usage Identifiers
- Version Compatibility with 1.0 Specifications
  - Media Types
  - Requesting formats
- Appendix A: Samples
- Appendix B: Resource Shapes
- Appendix C: Notices and References
  - Contributors
  - Reporting Issues on the Specification
  - Intellectual Property Covenant
License

This work is licensed under a Creative Commons Attribution License.

Notation and Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in RFC2119. Domain name examples use RFC2606.

Introduction

(this section is informative)

This specification defines a RESTful web services interface for Change Management, 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 Change Management working group 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 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.

The following figure illustrates how this CM specification relates to other OSLC specifications. It extends and restricts the OSLC Core, while referencing resources defined in other domain specifications.

Terminology

**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.

**Consumer** - an implementation of the OSLC Change Management specifications as a client. OSLC CM Consumers consume services provided by service providers

**Service Provider** - an implementation of the OSLC Change Management specifications as a server. OSLC CM clients consume these services

Base Requirements

Compliance

This specification is based on OSLC Core Specification. OSLC CM consumers and service providers **MUST** be compliant with both the core specification and this CM specification, and **SHOULD** follow all the guidelines and recommendations in both these specifications.

The following table summarizes the requirements from OSLC Core Specification as well as some additional specific to CM. Note that this specification further restricts some of the requirements for OSLC Core Specification. See further sections in this specification or the OSLC Core
<table>
<thead>
<tr>
<th>Requirement</th>
<th>Level</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown properties and content</td>
<td>MAY / MUST</td>
<td>OSLC services MAY ignore unknown content and OSLC clients MUST preserve unknown content</td>
</tr>
<tr>
<td>Resource Operations</td>
<td>MUST</td>
<td>OSLC service MUST support resource operations via standard HTTP operations</td>
</tr>
<tr>
<td>Resource Paging</td>
<td>MAY</td>
<td>OSLC services MAY provide paging for resources but only when specifically requested by client</td>
</tr>
<tr>
<td>Partial Resource Representations</td>
<td>MAY / MUST</td>
<td>OSLC services MUST support request for a subset of a resource’s properties via the oslc.properties URL parameter retrieval via HTTP GET and MAY support via HTTP PUT</td>
</tr>
<tr>
<td>Partial Update</td>
<td>MAY</td>
<td>OSLC services MAY support partial update of resources using patch semantics</td>
</tr>
<tr>
<td>Service Provider Resources</td>
<td>MAY / MUST</td>
<td>OSLC service providers MAY provide a Service Provider Catalog and MUST provide a Service Provider resource</td>
</tr>
<tr>
<td>Creation Factories</td>
<td>MUST</td>
<td>OSLC service providers MUST provide creation factories to enable resource creation via HTTP POST</td>
</tr>
<tr>
<td>Query Capabilities</td>
<td>MUST</td>
<td>OSLC service providers MUST provide query capabilities to enable clients to query for resources</td>
</tr>
<tr>
<td>Query Syntax</td>
<td>MUST</td>
<td>OSLC query capabilities MUST support the OSLC Core Query Syntax and MAY use other query syntax</td>
</tr>
<tr>
<td>Delegated UI Dialogs</td>
<td>MUST</td>
<td>OSLC Services MUST offer delegated UI dialogs (creation and selections) specified via service provider resource</td>
</tr>
<tr>
<td>UI Preview</td>
<td>SHOULD</td>
<td>OSLC Services SHOULD offer UI previews for resources that may be referenced by other resources</td>
</tr>
<tr>
<td>HTTP Basic Authentication</td>
<td>MAY</td>
<td>OSLC Services MAY support Basic Auth and should do so only over HTTPS</td>
</tr>
<tr>
<td>OAuth Authentication</td>
<td>SHOULD</td>
<td>OSLC Services SHOULD support OAuth and can indicate the required OAuth URLs via the service provider resource</td>
</tr>
<tr>
<td>Error Responses</td>
<td>MAY</td>
<td>OSLC Services MAY provide error responses using Core defined error formats</td>
</tr>
<tr>
<td>RDF/XML Representations</td>
<td>MUST / SHOULD</td>
<td>OSLC services MUST provide an RDF/XML representation for HTTP GET requests and SHOULD support RDF/XML representations on POST and PUT requests.</td>
</tr>
<tr>
<td>XML Representations</td>
<td>MUST</td>
<td>OSLC services MUST provide a XML representation for HTTP GET, POST and PUT requests that conform to the Core Guidelines for XML.</td>
</tr>
<tr>
<td>JSON Representations</td>
<td>MUST</td>
<td>OSLC services MUST provide JSON representations for HTTP GET, POST and PUT requests that conform to the Core Guidelines for JSON.</td>
</tr>
<tr>
<td>HTML Representations</td>
<td>SHOULD</td>
<td>OSLC services SHOULD provide HTML representations for HTTP GET requests</td>
</tr>
</tbody>
</table>

**Specification Versioning**

See [OSLC Core Specification Versioning section](#).

Service providers that support the resource formats and services in this specification **MUST** use HTTP response header of `OSLC-Core-Version` with a value of 2.0. Consumers **MAY** request formats and services defined in this document by providing a HTTP request header of `OSLC-Core-Version` with a value of 2.0. See section below on Version Compatibility with OSLC CM 1.0 Specifications.

**Namespaces**

In addition to the namespace URIs and namespace prefixes `oslc`, `rdf`, `dcterms` and `foaf` defined in the [OSLC Core specification](#), 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#` (Reference: OSLC RM)
- `oslc_qm`: `http://open-services.net/ns/qm#` (Reference: OSLC QM)
- `oslc_scm`: `http://open-services.net/ns/scm#` (Reference: OSLC SCM)

**Resource Formats**
In addition to the requirements for OSLC Core Resource Formats section, this section outlines further refinements and restrictions.

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

- CM Providers **MUST** provide RDF/XML, XML, and JSON representations. The XML and JSON representations **SHOULD** follow the guidelines outlined in the OSLC Core Representations Guidance.
- CM Consumers requesting RDF/XML **SHOULD** be prepared for any valid RDF/XML document. CM Consumers requesting XML or JSON **SHOULD** be prepared for representations that follow the guidelines outlined in the OSLC Core Representations Guidance.
- CM Providers **SHOULD** support an [X]HTML representation and a user interface (UI) preview as defined by UI Preview Guidance.

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

- CM Providers **MUST** accept XML and JSON representations and **SHOULD** accept RDF/XML representations. CM Providers accepting RDF/XML **SHOULD** be prepared for any valid RDF/XML document. For XML or JSON, CM Providers **SHOULD** be prepared for representations that follow the guidelines outlined in the OSLC Core Representations Guidance.

For HTTP GET response formats for Query requests,

CM Providers **MUST** provide RDF/XML, XML, Atom Syndication Format XML and JSON representations.

When CM Consumers request:

- application/rdf+xml CM Providers **MUST** respond with RDF/XML representation without restrictions.
- application/json CM Providers **MUST** respond with JSON representation as defined in the OSLC Core Representations Guidance.
- application/xml CM Provider **MUST** respond with OSLC-defined abbreviated XML representation as defined in the OSLC Core Representations Guidance.
- application/atom+xml CM Provider **MUST** 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.

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

### Content Negotiation

OSLC Core Guidance clearly points to RDF representations (and specifically RDF/XML) as a convention that all OSLC Provider implementations minimally provide and accept. OSLC CM Provider 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. The OSLC Core describes an algorithm for generating consistent formats that are used as examples only.

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

**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.

### Authentication

See OSLC Core Authentication section. In addition to the OSLC Core authentication requirements, OSLC CM services providers **SHOULD** support OAuth.

### Error Responses

See OSLC Core Error Responses section. OSLC CM puts no additional constraints on error responses.

### Pagination
OSLC CM service providers SHOULD support pagination of query results and MAY support pagination of a single resource’s properties as defined by the OSLC Core Specification.

**Requesting and Updating Properties**

**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 service provider 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.

**Updating a Subset of Properties**

A client MAY request that a subset of a resource's properties be updated by identifying those properties to be modified using the `oslc.properties` URL parameter on a HTTP PUT request.

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.

**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 Service Providers SHOULD provide support for a partial update of the multi-valued properties as defined by OSLC Core Partial Update.

**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. In queries the `oslc_cm:status` property is 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 service providers 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.

Predicates are exposed as single-value often read-only properties on a Change Request resource. 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.

**Labels for Relationships**

Change Management relationships to other resources are represented by RDF properties. Instances of a relationship - often called links - are RDF triples whose predicate is the property, and whose value (aka object) is the URI of target resource. When a Change Management 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. It is recommended to use either a property from the target resource such as `dcterms:title` or retrieve a label for presentation by retrieving the target resource's OSLCUIPreview. In the case where a relationship (a triple) requires a unique label that is not available from the target resource, only then OSLC providers MAY support a `dcterms:label` link property in Change Management resource representations, using the anchor approach outlined in the OSLC Core Links Guidance.

RDF/XML and XML example using reified statement:
CM Resource Definitions

Property value types that are not defined in the following sections, are defined in OSLC Core - Defining OSLC Properties

Resource ChangeRequest

The Change Request resource is a single definition used to define many kinds of change requests such as: defect, enhancement, task, bug, activity, etc. There are a fair number of common properties between these different kinds of change requests and can use some of the properties in the follow definition to identify them.

The Change Request resource properties are not limited to the ones defined in this specification, service providers may provide additional properties. It is recommended that any additional properties exist in their own unique namespace and not use the namespaces defined in these specifications.

- **Name**: ChangeRequest
- **Type URI**: http://open-services.net/ns/cm#ChangeRequest

<table>
<thead>
<tr>
<th>Prefixed Name</th>
<th>Occurs</th>
<th>Read-only</th>
<th>Value-type</th>
<th>Representation</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>oslc:shortTitle</td>
<td>zero-or-one</td>
<td>unspecified</td>
<td>XMLLiteral</td>
<td>n/a</td>
<td>n/a</td>
<td>Short name identifying a resource, often used as an abbreviated identifier for</td>
</tr>
<tr>
<td>Property</td>
<td>Cardinality</td>
<td>Validity</td>
<td>Type</td>
<td>Value</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>----------</td>
<td>------------</td>
<td>------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>dcterms:description</td>
<td>zero-or-one</td>
<td>unspecified</td>
<td>XMLLiteral</td>
<td>n/a</td>
<td>Descriptive text (reference: Dublin Core) about resource represented as rich text in XHTML content. SHOULD include only content that is valid and suitable inside an XHTML &lt;span&gt; element.</td>
<td></td>
</tr>
<tr>
<td>dcterms:title</td>
<td>exactly-one</td>
<td>unspecified</td>
<td>XMLLiteral</td>
<td>n/a</td>
<td>Title (reference: Dublin Core) or often a single line summary of the resource represented as rich text in XHTML content. SHOULD include only content that is valid and suitable inside an XHTML &lt;div&gt; element.</td>
<td></td>
</tr>
<tr>
<td>dcterms:identifier</td>
<td>exactly-one</td>
<td>True</td>
<td>String</td>
<td>n/a</td>
<td>A unique identifier for a resource. Assigned by the service provider when a resource is created. Not intended for end-user display.</td>
<td></td>
</tr>
<tr>
<td>dcterms:subject</td>
<td>zero-or-many</td>
<td>False</td>
<td>String</td>
<td>n/a</td>
<td>Tag or keyword for a resource. Each occurrence of a dcterms:subject property denotes an additional tag for the resource.</td>
<td></td>
</tr>
<tr>
<td>dcterms:creator</td>
<td>zero-or-many</td>
<td>unspecified</td>
<td>Either Resource or Local Resource</td>
<td>Either Reference or Inline</td>
<td>any</td>
<td>Creator or creators of resource (reference: Dublin Core). It is likely that the target resource will be a foaf:Person but that is not necessarily the case.</td>
</tr>
<tr>
<td>dcterms:contributor</td>
<td>zero-or-many</td>
<td>unspecified</td>
<td>Either Resource or Local Resource</td>
<td>Either Reference or Inline</td>
<td>any</td>
<td>The person(s) who are responsible for the work needed to complete the change request (reference: Dublin Core). It is likely that the target resource will be a foaf:Person but that is not necessarily the case.</td>
</tr>
<tr>
<td>dcterms:created</td>
<td>zero-or-one</td>
<td>True</td>
<td>DateTime</td>
<td>n/a</td>
<td>Timestamp of resource creation (reference: Dublin Core).</td>
<td></td>
</tr>
<tr>
<td>dcterms:modified</td>
<td>zero-or-one</td>
<td>True</td>
<td>DateTime</td>
<td>n/a</td>
<td>Timestamp last latest resource modification (reference: Dublin Core).</td>
<td></td>
</tr>
<tr>
<td>rdf:type</td>
<td>zero-or-many</td>
<td>unspecified</td>
<td>Resource</td>
<td>Reference</td>
<td>n/a</td>
<td>The resource type URIs. One of at least has the value <a href="http://open-services.net/ns/cm#ChangeRequest">http://open-services.net/ns/cm#ChangeRequest</a></td>
</tr>
<tr>
<td>oslc:serviceProvider</td>
<td>zero-or-many</td>
<td>unspecified</td>
<td>Resource</td>
<td>Reference</td>
<td>oslc:ServiceProvider</td>
<td>The scope of a resource is a URI for the resource's OSLC Service Provider.</td>
</tr>
<tr>
<td>oslc:instanceShape</td>
<td>zero-or-one</td>
<td>unspecified</td>
<td>Resource</td>
<td>Reference</td>
<td>oslc:ResourceShape</td>
<td>Resource Shape that provides hints as to resource property value-types and allowed values.</td>
</tr>
<tr>
<td>oslc:discussedBy</td>
<td>zero-or-one</td>
<td>unspecified</td>
<td>Resource</td>
<td>Either</td>
<td>oslc:Discussion</td>
<td>A series of notes and comments about this change request.</td>
</tr>
</tbody>
</table>
### OSLC CM: Start of additional properties

<table>
<thead>
<tr>
<th>Prefixed Name</th>
<th>Occurs</th>
<th>Read-only</th>
<th>Value-type</th>
<th>Representation</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>oslc_cm:closeDate</td>
<td>zero-or-one</td>
<td>true</td>
<td>DateTime</td>
<td>n/a</td>
<td>n/a</td>
<td>The date at which no further activity or work is intended to be conducted.</td>
</tr>
<tr>
<td>oslc_cm:status</td>
<td>zero-or-one</td>
<td>unspecified</td>
<td>String</td>
<td>n/a</td>
<td>n/a</td>
<td>Used to indicate the status of the change request based on values defined by the service provider. Most often a read-only property. Some possible values may include: 'Submitted', 'Done', 'InProgress', etc.</td>
</tr>
</tbody>
</table>

### State predicate properties: This grouping of properties define a set of computed state predicates, see section on State Predicates for more information. The only restriction on valid state predicate combinations is that if oslc_cm:inprogress is true, then oslc_cm:fixed and oslc_cm:closed must also be false

<table>
<thead>
<tr>
<th>Prefixed Name</th>
<th>Occurs</th>
<th>Read-only</th>
<th>Value-type</th>
<th>Representation</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>oslc_cm:closed</td>
<td>zero-or-one</td>
<td>True</td>
<td>Boolean</td>
<td>n/a</td>
<td>n/a</td>
<td>Whether or not the Change Request is completely done, no further fixes or fix verification is needed.</td>
</tr>
<tr>
<td>oslc_cm:inprogress</td>
<td>zero-or-one</td>
<td>True</td>
<td>Boolean</td>
<td>n/a</td>
<td>n/a</td>
<td>Whether or not the Change Request in a state indicating that active work is occurring. If oslc_cm:inprogress is true, then oslc_cm:fixed and oslc_cm:closed must also be false</td>
</tr>
<tr>
<td>oslc_cm:fixed</td>
<td>zero-or-one</td>
<td>True</td>
<td>Boolean</td>
<td>n/a</td>
<td>n/a</td>
<td>Whether or not the Change Request has been fixed.</td>
</tr>
<tr>
<td>oslc_cm:approved</td>
<td>zero-or-one</td>
<td>True</td>
<td>Boolean</td>
<td>n/a</td>
<td>n/a</td>
<td>Whether or not the Change Request has been approved.</td>
</tr>
<tr>
<td>oslc_cm:reviewed</td>
<td>zero-or-one</td>
<td>True</td>
<td>Boolean</td>
<td>n/a</td>
<td>n/a</td>
<td>Whether or not the Change Request has been reviewed.</td>
</tr>
<tr>
<td>oslc_cm:verified</td>
<td>zero-or-one</td>
<td>True</td>
<td>Boolean</td>
<td>n/a</td>
<td>n/a</td>
<td>Whether or not the resolution or fix of the Change Request has been verified.</td>
</tr>
</tbody>
</table>

### Relationship properties: This grouping of properties are used to identify relationships between resources managed by other OSLC Service Providers

<table>
<thead>
<tr>
<th>Prefixed Name</th>
<th>Occurs</th>
<th>Read-only</th>
<th>Value-type</th>
<th>Representation</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>oslc_cm:relatedChangeRequest</td>
<td>zero-or-many</td>
<td>False</td>
<td>Resource</td>
<td>Reference</td>
<td>any</td>
<td>This relationship is loosely coupled and has no specific meaning. It is likely that the target resource will be an oslc_cm:ChangeRequest but that is not necessarily the case.</td>
</tr>
<tr>
<td>oslc_cm:affectsPlanItem</td>
<td>zero-or-many</td>
<td>False</td>
<td>Resource</td>
<td>Reference</td>
<td>any</td>
<td>Change request affects a plan item. It is likely that the target resource will be an oslc_cm:ChangeRequest but that is not necessarily the case.</td>
</tr>
<tr>
<td>oslc_cm:affectedByDefect</td>
<td>zero-or-many</td>
<td>False</td>
<td>Resource</td>
<td>Reference</td>
<td>any</td>
<td>Change request is affected by a</td>
</tr>
</tbody>
</table>
or-
many

<table>
<thead>
<tr>
<th>Property</th>
<th>Multiplicity</th>
<th>Type</th>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
</table>
| oslc_cm:tracksRequirement    | zero-or-
many | False         | Resource    | Reference any Tracks the associated Requirement or Requirement ChangeSet resources. It is likely that the target resource will be an oslc_cm:ChangeRequest but that is not necessarily the case. |
| oslc_cm:implementsRequirement| zero-or-
many | False         | Resource    | Reference any Implements associated Requirement. It is likely that the target resource will be an oslc_rm:Requirement but that is not necessarily the case. |
| oslc_cm:affectsRequirement   | zero-or-
many | False         | Resource    | Reference any Change request affecting a Requirement. It is likely that the target resource will be an oslc_rm:Requirement but that is not necessarily the case. |
| oslc_cm:tracksChangeSet      | zero-or-
many | False         | Resource    | Reference any Tracks SCM change set resource. It is likely that the target resource will be an oslc_scm:ChangeSet but that is not necessarily the case. |

Naming convention for relationship properties follows this pattern:

- **related** - Identifies a loose relationship between a Change Request and referenced resource. These relationships can be used to name associated resources managed by other service providers.
- **tracks** - Identifies that a Change Request is used to track the lifecycle of referenced resource. From the CM tool perspective, these relationships can be used to track work that needs to be done for referenced resources.
- **affects** - Indicates that the Change Request affects, has been predetermined to have impact, related resource. These property relationships can be used to understand the potential impact of referenced resources.

### Deprecated terms

Since the initial development of the OSLC CM 2.0 specification, there have been a number of properties that are no longer recommended for use and therefore marked as deprecated. Details about the reason for being marked as such and backwards compatibility items, refer to the [CM 2.0 Issues page](#).
<table>
<thead>
<tr>
<th>Prefixed Name</th>
<th>Occurs</th>
<th>Read-only</th>
<th>Value-type</th>
<th>Representation</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deprecated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A short string representation for the type, example 'Defect'.</td>
</tr>
<tr>
<td>dcterms:type</td>
<td>zero-or-more</td>
<td></td>
<td>unspecified</td>
<td>String</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>oslc_cm:testedByTestCase</td>
<td>zero-or-many</td>
<td>False</td>
<td>Resource</td>
<td>Reference</td>
<td>any</td>
<td>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.</td>
</tr>
<tr>
<td>oslc_cm:affectsTestResult</td>
<td>zero-or-many</td>
<td>False</td>
<td>Resource</td>
<td>Reference</td>
<td>any</td>
<td>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.</td>
</tr>
<tr>
<td>oslc_cm:blocksTestExecutionRecord</td>
<td>zero-or-many</td>
<td>False</td>
<td>Resource</td>
<td>Reference</td>
<td>any</td>
<td>Associated QM resource that is blocked by this Change Request. It is likely that the target resource will be an oslc_qm:TestExecutionRecord but that is not necessarily the case.</td>
</tr>
<tr>
<td>oslc_cm:relatedTestExecutionRecord</td>
<td>zero-or-many</td>
<td>False</td>
<td>Resource</td>
<td>Reference</td>
<td>any</td>
<td>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.</td>
</tr>
<tr>
<td>oslc_cm:relatedTestCase</td>
<td>zero-or-many</td>
<td>False</td>
<td>Resource</td>
<td>Reference</td>
<td>any</td>
<td>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.</td>
</tr>
<tr>
<td>oslc_cm:relatedTestPlan</td>
<td>zero-or-many</td>
<td>False</td>
<td>Resource</td>
<td>Reference</td>
<td>any</td>
<td>Related QM test plan resource. It is likely that the target resource will be an oslc_qm:TestPlan but that is not necessarily the case.</td>
</tr>
<tr>
<td>oslc_cm:relatedTestScript</td>
<td>zero-or-many</td>
<td>False</td>
<td>Resource</td>
<td>Reference</td>
<td>any</td>
<td>Related QM test script resource. It is likely that the target resource will be an oslc_qm:TestScript but that is not necessarily the case.</td>
</tr>
</tbody>
</table>

**CM Service Provider Capabilities**

**Service Provider Resources**

OSLC CM service providers **MUST** provide a Service Provider Resource that can be retrieved at a implementation dependent URI.

OSLC CM service providers **MAY** provide a Service Provider Catalog Resource that can be retrieved at a implementation dependent URI.

OSLC CM service providers **MAY** provide a oslc:serviceProvider property for their defined resources that will be the URI to a Service Provider Resource.

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

**Creation Factories**

OSLC CM service providers **MUST** support Creation Factories and list them in the Service Provider Resource as defined by OSLC Core. OSLC CM service providers **SHOULD** support Resource Shapes for Creation Factories as defined in OSLC Core Specification.
OSLC CM service providers MUST support the Query Capabilities as defined by OSLC Core. OSLC CM service providers SHOULD support Resource Shapes for Query Capability as defined in OSLC Core Specification.

The Query Capability MUST support these parameters:

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

If shape information is NOT present with the Query Capability, service providers 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

### Delegated UIs

OSLC CM service providers MUST support the selection and creation of resources by delegated web-based user interface dialogs Delegated UIs as defined by OSLC Core.

OSLC CM service providers MAY support the pre-filling of creation dialogs based on the definition at Delegated UIs.

### Usage Identifiers

OSLC CM service provider can identify 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 consumers when multiple entries are found.

The additional 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

### Version Compatibility with 1.0 Specifications

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

Network addressable resource URIs used for 1.0 resources for these types: ChangeRequest, ServiceDescriptor and ServiceProviderCatalog, should not have to change. Consumers who support both 1.0 and 2.0, should only preserve these resource URIs. When a Provider 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 UIs, factories, etc).

### Media Types

For a Change Request Resource format identification of RDF/XML and XML, the media type used for this representation SHOULD be application/rdf+xml or application/xml. The usage of the OSLC CM 1.0 defined media types of application/x-oslc-cm-change-request+xml, application/x-oslc-cm-service-description+xml and application/x-oslc-disc-service-provider-catalog+xml is being deprecated.

For a Change Request Resource format identification of JSON, the media type used for this representation SHOULD be application/json. The usage of the OSLC CM 1.0 defined media type of application/x-oslc-cm-change-request+json is being deprecated.

### Requesting formats
CM 1.0 consumers wanting to request 1.0 resource formats will not need to change if they used 1.0 defined media types (application/x-oslc-cm*), see OSLC-CM 1.0. CM 2.0 consumers should use media types as defined in this specification for requests, excluding the OSLC CM 1.0 specific media types (application/x-oslc-cm*). CM consumers supporting both 1.0 and 2.0, should request both 1.0 and 2.0 media types on HTTP GET requests as usually done with HTTP request parameter `Accept` giving appropriate quality (See HTTP Accept) weighting to help distinguish their preferred content.

For additional guidance, a CM 2.0 consumer or provider may reference the OSLC-Core-Version HTTP header with a value of 2.0.

**Appendix A: Samples**

(this section is informative)

See CmSpecificationV2Samples

**Appendix B: Resource Shapes**

(this section is informative)

See CmSpecificationV2Shapes

**Appendix C: Notices and References**

**Contributors**

- SteveSpeicher (IBM, OSLC-CM Lead)
- OlivierBerger (Institut TELECOM)
- ScottBosworth (IBM)
- GaryDang (Accenture)
- SusanDuncan (Oracle)
- RobertElves (Tasktop)
- MatsGothe (IBM)
- DaveJohnson (IBM)
- MikKersten (Tasktop)
- SamLee (IBM)
- MarkRinger (Rally)
- EricSink (SourceGear)
- PatrickStreule (IBM)
- MattThomas (BSD Group)
- DenisTyrell (Oracle)
- RandyVogel (Accenture)
- AndreWeinand (IBM)
- SofiaYeung (Oracle)

**Reporting Issues on the Specification**

The working group participants who author and maintain this working draft specification, monitor a distribution list where issues or questions can be raised, see Change Management Mailing List

Also the issues found with this specification and their resolution can be found at CmSpecV2Issues

**Intellectual Property Covenant**

The members of the Working Group (or as appropriate, their employers) have documented a Patent Non-Assertion Covenant for implementations of the CM 2.0 Specification, as described in the open-services.net Terms of Use. Details of the Covenant may be found here.

**References**