

# OSLC Core Version 3.0. Part 7: Vocabulary

Committee Specification Draft 01 /  
Public Review Draft 01

07 June 2016

## Specification URIs

### This version:

<http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd01/part7-core-vocabulary/oslc-core-v3.0-csprd01-part7-core-vocabulary.html>  
<http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd01/part7-core-vocabulary/oslc-core-v3.0-csprd01-part7-core-vocabulary.pdf>

### Previous version:

N/A

### Latest version:

<http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/oslc-core-v3.0-part7-core-vocabulary.html>  
<http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/oslc-core-v3.0-part7-core-vocabulary.pdf>

### Technical Committee:

[OASIS OSLC Lifecycle Integration Core \(OSLC Core\) TC](#)

### Chairs:

[Jim Amsden \(jamsden@us.ibm.com\)](mailto:jamsden@us.ibm.com), IBM  
[Martin Sarabura \(msarabura@ptc.com\)](mailto:msarabura@ptc.com), PTC

### Editor:

[Jim Amsden \(jamsden@us.ibm.com\)](mailto:jamsden@us.ibm.com), IBM

### Additional artifacts:

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

- OSLC Core Version 3.0. Part 1: Overview. <http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd01/part1-overview/oslc-core-v3.0-csprd01-part1-overview.html>
- OSLC Core Version 3.0. Part 2: Discovery. <http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd01/part2-discovery/oslc-core-v3.0-csprd01-part2-discovery.html>
- OSLC Core Version 3.0. Part 3: Resource Preview. <http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd01/part3-resource-preview/oslc-core-v3.0-csprd01-part3-resource-preview.html>
- OSLC Core Version 3.0. Part 4: Delegated Dialogs. <http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd01/part4-delegated-dialogs/oslc-core-v3.0-csprd01-part4-delegated-dialogs.html>
- OSLC Core Version 3.0. Part 5: Attachments. <http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd01/part5-attachments/oslc-core-v3.0-csprd01-part5-attachments.html>
- OSLC Core Version 3.0. Part 6: Resource Shape. <http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd01/part6-resource-shape/oslc-core-v3.0-csprd01-part6-resource-shape.html>
- OSLC Core Version 3.0. Part 7: Vocabulary (this document). <http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd01/part7-core-vocabulary/oslc-core-v3.0-csprd01-part7-core-vocabulary.html>

### Related work:

This specification is related to:

- OSLC Core Version 3.0: Link Guidance. Work in progress. Current draft: <https://tools.oasis-open.org/version-control/svn/oslc-core/trunk/supporting-docs/link-guidance.html>

### RDF Namespaces:

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

## Abstract:

Core Vocabulary defines the OSLC Core RDF vocabulary terms and resources, that have broad applicability across various domains.

## Status:

This document was last revised or approved by the [OASIS OSLC Lifecycle Integration Core \(OSLC Core\) TC](#) on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at <https://www.oasis-open.org/committees/oslc-core#technical>.

TC members should send comments on this specification to the TC's email list. Others should send comments to the TC's public comment list [oslc-core-comments@lists.oasis-open.org](mailto:oslc-core-comments@lists.oasis-open.org), after subscribing to it by following the instructions at the "Send A Comment" button on the TC's web page at <https://www.oasis-open.org/committees/oslc-core/>.

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 Intellectual Property Rights section of the TC's web page (<https://www.oasis-open.org/committees/oslc-core/ipr.php>).

## Citation format:

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

### [OSLC-CoreVocab-3.0]

*OSLC Core Version 3.0. Part 7: Vocabulary*. Edited by Jim Amsden. 07 June 2016. OASIS Committee Specification Draft 01 / Public Review Draft 01. <http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd01/part7-core-vocabulary/oslc-core-v3.0-csprd01-part7-core-vocabulary.html>. Latest version: <http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/oslc-core-v3.0-part7-core-vocabulary.html>.

---

## Notices

Copyright © OASIS Open © 2016. 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 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 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 Committee Specification or OASIS Standard, to notify 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 Technical Committee 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 Committee 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. Motivation
  - 3. Resource Shape
  - 4. Defining Enumerations
  - 5. Common Properties
    - 5.1 Properties on Any Resource
    - 5.2 Person Properties
    - 5.3 Implementation Conformance
  - 6. Discussion
    - 6.1 Shape: Discussion
    - 6.2 Shape: Comment
  - 7. Errors
    - 7.1 Implementation Conformance
    - 7.2 Shape: Error
    - 7.3 Shape: ExtendedError
  - 8. Terms for describing vocabularies
    - 8.1 Inverse Labels
    - 8.2 Traceability and Impact type
  - 9. Discovery
  - 10. Terms
    - 10.1 Vocabulary Details
- Appendix A. Change History

---

## 1. Introduction

*This section is non-normative.*

Various resources and properties may be so commonly used or apply so broadly that it makes sense to define them in one place so they can be easily reused. Some common examples are short names or labels, error messages, discussion threads, traceability/impacts relationship behavior or annotating other vocabulary terms.

### 1.1 Terminology

Terminology uses and extends the terminology and capabilities of OSLC Core Overview [OSLCCore3], W3C Linked Data Platform [LDP], W3C's Architecture of the World Wide Web [WEBARCH], Hyper-text Transfer Protocol [HTTP11].

#### **Archived Resource**

A resource in which an explicit action has been performed to mark the resource as no longer active and may be removed from typical user interactions. As a consequence, an archived resource should be considered immutable.

### 1.2 References

#### 1.2.1 Normative references

[DC-TERMS]

Dublin Core Metadata Initiative. *Dublin Core Metadata Initiative Terms, version 1.1*. 11 October 2010. DCMI Recommendation. URL: <http://dublincore.org/documents/2010/10/11/dcmi-terms/>.

[FOAF]

Dan Brickley; Libby Miller. *FOAF Vocabulary Specification 0.99 (Paddington Edition)*. 14 January 2014. URL: <http://xmllns.com/foaf/spec>

[HTTP11]

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

[LDP]

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

[OSLCCore3]

Steve Speicher. *OSLC Core 3.0*. URL: <http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd01/part1-overview/oslc-core-v3.0-csprd01-part1-overview.html>

[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>

[ResourceShapes]

Arthur Ryman; Jim Amsden. *OSLC Resource Shape 3.0*. URL: <http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd01/part6-resource-shape/oslc-core-v3.0-csprd01-part6-resource-shape.html>

[rdf-schema]

Dan Brickley; Ramanathan Guha. *RDF Schema 1.1*. 25 February 2014. W3C Recommendation. URL: <http://www.w3.org/TR/rdf-schema/>

[rdf11-concepts]

Richard Cyganiak; David Wood; Markus Lanthaler. *RDF 1.1 Concepts and Abstract Syntax*. 25 February 2014. W3C Recommendation. URL: <http://www.w3.org/TR/rdf11-concepts/>

## 1.2.2 Informative references

[LinkGuidance]

Steve Speicher; Jim Amsden. *OSLC Link Guidance 3.0*. URL: <https://tools.oasis-open.org/version-control/svn/oslc-core/trunk/supporting-docs/link-guidance.html>

[SHACL]

Holger Knublauch; Arthur Ryman. *Shapes Constraint Language (SHACL)*. Draft. URL: <https://w3c.github.io/data-shapes/shacl/>

[WEBARCH]

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

[skos-reference]

Alistair Miles; Sean Bechhofer. *SKOS Simple Knowledge Organization System Reference*. 18 August 2009. W3C Recommendation. URL: <http://www.w3.org/TR/skos-reference>

## 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. Motivation

*This section is non-normative.*

Most OSLC vocabularies and resource shape constraints on usages of those vocabularies are given in the various OSLC domain specifications. The motivation for these domain specifications is to define agreed upon, formalized vocabulary terms for key elements in the domain. Domain vocabularies are not intended to restrict what vocabularies servers actually use for those domains, or what domains they support. Rather vocabularies establish a common core of domain terms that reduce accidental variability and foster greater interchange and interoperability between tools that support and users that make use of domains. Servers are free to extend the domains and integrate across domains as required to realize their provided capabilities.

OSLC Core takes a similar approach for common terms that are used across most domains. The intent is to provide a foundation for building domains that again reduces unnecessary variability, and eliminates the need for the various domain specifications to redundantly define similar terms. The follow paragraphs describe the kinds of common terms defined by OSLC core in order to achieve the stated intent.

Archived Resources are typically found in large systems in which an immutable copy of the state of a resource at a given time is captured. The purpose may vary in that it could be simply a way to facilitate access to a backup or snapshot of a resource at a particular point in time. Another use may be to indicate that a resource has been deleted, but is saved by the system for historical or legal reasons. Having a consistent way to indicate that a resource, or a set of them, has been archived helps when defining certain views of the resources or queries. Archived Resources may be identified by having a property `oslc:archived`, with value `true`.

Many different kinds of applications have a way to provide comments or notes related to a given resource. These take the form of a discussion, with a sequence of comments. OSLC Core provide a common way for applications to easily add to a comment to a discussion thread or navigate a discussion thread.

Error responses from HTTP request often take the form of HTML pages intended for a human to read, even though these requests are often initiated from applications that don't have a human actively monitoring it. OSLC Core defines a consistent way to request error responses of a certain format, and a prescribed interaction model that helps clients better handle errors automatically.

Consider a user interface for a query builder that allows users to build queries about test cases. It is natural for the query builder to present the user with a list of the properties that apply to test cases that could be used in the query. Suppose the user wants to build a query that returns all the requirements that are validated by a test case. The query builder should describe the available properties from the point of view of the test case. This implies that the query builder should describe the inverse relation asserted by any triple that has the test case as an object. In this example, the query builder should describe assertions of the form {requirement `oslc_rm:validatedBy` test case} as {test case `validates` requirement}.

Some RDF properties express relations or links between subject and object artifacts. If a change in state of subject and/or object of a triple may result in the assertion becoming invalid, the link may be seen to represent a dependency. OSLC Core provides property `oslc:impactType` as a means of defining the dependency represented by an RDF property.

## 3. Resource Shape

The shape of an RDF resource is a description of the set of triples it is expected to contain and the integrity constraints those triples are required to satisfy. Applications of shapes include validating RDF data, documenting RDF APIs, and providing meta-data to tools, such as form and query builders, that handle RDF data. OSLC Core uses shapes to:

- Define specific vocabulary constraints including allowed values, max size, cardinality, representation in RDF specifications and if the property is read only.  
Specify the properties required for resource creation.
- Specify what servers all allow for prefilling delegated dialogs.
- Describe the results of query operations.

Constraints on OSLC Core and Domain resources **SHOULD** be described using [ResourceShapes] which is included as part of the OSLC Core multi-part specifications. Servers **MAY** use other constraint languages such as [SHACL] to define resource constraints.

## 4. Defining Enumerations

*This section is non-normative.*

Some property values are characterized by a limited set of enumerated values. The type for these property values is called an enumeration in many modeling and programming languages, while the values are called enumeration literals. RDF does not define a specific way of defining enumerated types and enumeration literals. As a result, different vocabularies may take different, but equally valid approaches. In order to foster interoperability and integration, OSLC Core provides a recommended approach for defining enumerated types and enumeration literals. This approach is used in defining the OSLC Core vocabulary terms.

Enumerations in an OSLC vocabulary should be defined as an RDF class. Enumeration literals are the URIs of individuals of that class. For example, consider an enumeration called "Color" that has enumeration literals {red, yellow, green, blue} (using Java notation). Color would be defined as an RDF class and the enumeration literals would be individuals of that class. A color property is defined and then used to assert that the color of myCar is blue.

### EXAMPLE 1

```
# Color enumeration
Color
  a rdfs:Class ;
  rdfs:label "Color" ;
  rdfs:comment "The class of possible color values." .

# Color enumeration literals
red
  a Color ;
  rdfs:label "red" .

yellow
  a Color ;
  rdfs:label "yellow" .

green
  a Color ;
  rdfs:label "green" .

blue
  a Color ;
  rdfs:label "blue" .

# A Color property
color
  a rdf:Property ;
  rdfs:label "color" ;
  rdfs:comment "Used to specify the color of a resource".

# Asserting the color of a resource
myCar color blue.
```

Enumerations can be open or closed. Open enumerations allow additional enumeration literals to be added as needed. Closed enumerations have a fixed set of enumeration literals that is not intended to be extended. Resource shapes can be used to constrain enumerations to a specific set of values. Notice in the example above that the color property did not specify its rdfs:range. This keeps the enumeration completely open to any set of individuals. OSLC prefers to use resource shapes to constrain resources for particular usages, leaving them open for extension for other, possibly unanticipated usages.

A shape can be used to constrain the Color enumeration for a specific purpose. For example, the color of lights in a traffic light should be constrained to exactly red, yellow and green.

### EXAMPLE 2

```
# Create a constraint on Color for traffic lights
TrafficLightConstraint
  a oslc:ResourceShape ;
  oslc:describes fhwa:TrafficLight ;
  dcterms:title "Establish constraints for traffic light colors" ;
  oslc:property colorConstraint .

colorConstraint
  a oslc:Property ;
  oslc:name "color" ;
  dcterms:description "The colors for a traffic light as specified by FHWA."
  oslc:propertyDefinition color ;
  oslc:occurs oslc:Exactly-one ;
  oslc:range Color ;
  oslc:allowValue red, yellow, green ;
  oslc:readOnly "false" ;
  oslc:representation oslc:Reference ;
  oslc:valueType oslc:Resource .
```

TrafficLightConstraint defines a constraint associated with the vocabulary term `fhwa:TrafficLight`. The constraint has one property, `colorConstraint` whose `oslc:propertyDefinition` is the `color` RDF property. The `oslc:range` for the `colorConstraint` is set to `Color`, meaning the value of the applicable property is constrained to be of `rdf:type Color`. The `oslc:allowedValue` property further constrains the values to be red, yellow, or green. If the `oslc:allowedValue` were not specified, then the `TrafficLightConstraint` would allow the enumeration to be open.

A completely different shape constraint could be used for colors that represent the status of a risk mitigation in a software development project.

## 5. Common Properties

Unlike the rest of the Core specification, these properties change and grow as new common properties are added by the Core TC. The properties that we list here are available for use in OSLC domain specifications when defining OSLC resources, but this does not mean that they are required to be in OSLC resources. OSLC domain specifications decide which properties are allowed and required for resources needed to realize their use cases. The OSLC common properties include properties defined in other standard vocabularies including:

- [Friend of a Friend \(FOAF\)](#)
- [Dublin Core \(dcterms\)](#)
- [RDF Schema \(rdfs\)](#)

### 5.1 Properties on Any Resource

Defines common properties that may be applicable to any OSLC resource. OSLC domains **SHOULD** use these properties where applicable rather than defining their own properties. The cardinality, representations, ranges, and other columns of the following table indicate typical usage, but a domain is free to apply its own constraints for particular resource shapes.

Prefixed Name	Occurs	Read-only	Value-type	Representation	Range	Description
<code>dcterms:contributor</code>	Zero-or-many	unspecified	AnyResource	Either	<code>foaf:Person</code>	Contributor or contributors to the resource. It is likely that the target resource will be a <code>foaf:Person</code> but that is not necessarily the case.
<code>dcterms:created</code>	Zero-or-one	unspecified	dateTime	N/A	Unspecified	Timestamp of resource creation
<code>dcterms:creator</code>	Zero-or-many	unspecified	AnyResource	Either	<code>foaf:Person</code>	Creator or creators of the resource. It is likely that the target resource will be a <code>foaf:Person</code> but that is not necessarily the case.
<code>dcterms:description</code>	Zero-or-many	unspecified	XMLLiteral	N/A	Unspecified	Descriptive text about resource represented as rich text in XHTML content.
<code>dcterms:identifier</code>	Zero-or-many	unspecified	string	N/A	Unspecified	A unique identifier for a resource. Typically read-only and assigned by the service provider when a resource is created. Not typically intended for end-user display.
<code>dcterms:modified</code>	Zero-or-many	unspecified	dateTime	N/A	Unspecified	Timestamp of latest resource modification.
<code>dcterms:relation</code>	Zero-or-many	unspecified	AnyResource	Either	Unspecified	Relation which identifies a related

<code>dcterms:subject</code>	Zero-or-many	unspecified	string	N/A	Unspecified	resource. Tag or keyword for a resource. Each occurrence of a <code>dcterms:subject</code> property denotes an additional tag for the resource.
<code>dcterms:title</code>	Zero-or-many	unspecified	XMLLiteral	N/A	Unspecified	Title of the resource represented as rich text in XHTML content.
<code>oslc:discussedBy</code>	Zero-or-many	unspecified	Resource	Either	<code>oslc:Comment</code>	A series of notes and comments about this resource.
<code>oslc:instanceShape</code>	Zero-or-many	unspecified	Resource	Reference	<code>oslc:ResourceShape</code>	The URI of a Resource Shape that describes the possible properties, occurrence, value types, allowed values and labels. This shape information is useful in displaying the subject resource as well as guiding clients in performing modifications. Instance shapes may be specific to the authenticated user associated with the request that retrieved the resource, the current state of the resource and other factors and thus should not be cached.
<code>oslc:modifiedBy</code>	Zero-or-many	unspecified	Resource	Either	<code>foaf:Person</code>	The URI of a resource describing the entity that most recently modified the subject resource. The link target is usually a <code>foaf:Person</code> or <code>foaf:Agent</code> , but could be any type. This is modeled after <code>dcterms:creator</code> , but Dublin Core currently has no equivalent property. A link to the resource's OSLC Service

<code>oslc:serviceProvider</code>	Zero-or-many	unspecified	Resource	Reference	<code>oslc:ServiceProvider</code>	Provider. There may be cases when the subject resource is available from a service provider that implements multiple domain specifications, which could result in multiple values for this property.
<code>oslc:shortId</code>	Zero-or-many	unspecified	string	N/A	Unspecified	A short, human-readable, plain text value. This value should be unique in some context that is apparent to human users of a service.
<code>oslc:shortTitle</code>	Zero-or-many	unspecified	XMLLiteral	N/A	Unspecified	Shorter form of <code>dcterms:title</code> for the resource represented as rich text in XHTML content.
<code>rdf:type</code>	Zero-or-many	unspecified	Resource	Reference	<code>rdfs:Class</code>	The resource type URIs.
<code>rdfs:member</code>	Zero-or-many	unspecified	Resource	Either	Unspecified	OSLC domains might define a number of member or contains relationships between resources. The <code>rdfs:member</code> property is suitable for use when only one such relationship needs to be defined, or when no additional semantics need to be implied by the property name

## 5.2 Person Properties

- **Name:** `Person`
- **URI:** `http://xmlns.com/foaf/0.1/Person`
- **Summary:** Person is a resource defined by FOAF that is used as the value for a `dcterms:creator` or `dcterms:contributor` property. This shape specifies the recommended minimal FOAF Person properties that should be provided for OSLC.

### Person Properties

Prefix Name	Occurs	Read-only	Value-type	Representation	Range	Description
<code>foaf:familyName</code>	Zero-or-many	unspecified	string	N/A	Unspecified	Family name of person expressed as simple text string.
<code>foaf:givenName</code>	Zero-or-many	unspecified	string	N/A	Unspecified	Given name of person expressed as simple text string.
<code>foaf:mbox</code>	Zero-or-many	unspecified	string	N/A	Unspecified	A personal mailbox for this person, typically identified using the <code>mailto:</code> URI scheme (see RFC 2368).
<code>foaf:name</code>	Zero-or-many	unspecified	string	N/A	Unspecified	The full name of a person expressed as simple text string.

foaf:nick	Zero-or-many	unspecified	string	N/A	Unspecified	A short informal nickname or login identifier expressed as simple text string.
-----------	--------------	-------------	--------	-----	-------------	--

## 5.3 Implementation Conformance

5.3.1 Changes to the OSLC Core Vocabulary **MUST** be approved by the OASIS OSLC Core TC. The OSLC Core Vocabulary is assigned the namespace URI of the <http://open-services.net/ns/core#>.

5.3.2 Domain TCs and other extensions **MUST** contribute their vocabulary terms in a namespace which is assigned to them as an authority.

5.3.3 OSLC Core, domain and other extensions **SHOULD** reuse existing vocabulary terms from stable vocabularies such as [DC-TERMS], RDF [rdf11-concepts], RDF Schema [rdf-schema], [FOAF], [skos-reference] and OSLC. New vocabulary terms should only be created when there is no clear existing choice available. See the [LDP] [similar clause on reuse](#).

See [section 10. Terms](#) for details on common property terms.

## 6. Discussion

### 6.1 Shape: Discussion

It is common to collect a series of comments on a lifecycle resource, often referred to as a discussion. For example: tasks, bug reports, requirements, assets and so on, are often collected across various types of resources such as project. A project might reflect the planning of work to deliver a product that realizes the requirements as validated through test cases and bug reports. Discussions allow users to collaborate with each other for more efficient and effective delivery. This Discussion resource definition provides a minimal shape describing the needed properties.

- **Name:** `core#Discussion`
- **URI:** <http://open-services.net/ns/core#Discussion>
- **Summary:** OSLC Core Discussion Shape

#### core#Discussion Properties

Prefixed Name	Occurs	Read-only	Value-type	Representation	Range	Description
<code>oslc:comment</code>	Zero-or-many	false	AnyResource	Either	<code>oslc:Comment</code>	Comment about resource
<code>oslc:discussionAbout</code>	Exactly-one	false	Resource	Reference	Unspecified	Reference to associated resource

### 6.2 Shape: Comment

Used in conjunction with [Shape: Discussion](#) to provide a minimal resource definition for a collection of comments.

- **Name:** `core#Comment`
- **URI:** <http://open-services.net/ns/core#Comment>
- **Summary:** OSLC Core Comment Shape

#### core#Comment Properties

Prefixed Name	Occurs	Read-only	Value-type	Representation	Range	Description
<code>dcterms:created</code>	Exactly-one	unspecified	dateTime	N/A	Unspecified	When the comment resource was created.
<code>dcterms:creator</code>	Exactly-one	unspecified	AnyResource	Either	<code>foaf:Person</code>	The person who created the comment.
<code>dcterms:description</code>	Exactly-one	unspecified	XMLLiteral	N/A	Unspecified	Details or body of the comment. <b>SHOULD</b> include only content that is valid and suitable inside an XHTML <code>&lt;div&gt;</code> element.
<code>dcterms:identifier</code>	Exactly-one	unspecified	string	N/A	Unspecified	A service defined identifier
<code>dcterms:title</code>	Zero-or-one	unspecified	XMLLiteral	N/A	Unspecified	A brief title for the comment. <b>SHOULD</b> include only content that is valid and suitable inside an XHTML <code>&lt;span&gt;</code> element.
<code>oslc:inReplyTo</code>	Zero-or-one	unspecified	Resource	Reference	<code>oslc:Comment</code>	Reference to the comment to which this comment replies.

See [section 10. Terms](#) for details on discussion property terms.

## 7. Errors

### 7.1 Implementation Conformance

7.1.1 When an OSLC Server incurs an error, it is **RECOMMENDED** that useful information be provided to clients in the body of the HTTP response.

7.1.2 OSLC Servers **SHOULD** use the [Error resource](#) defined below as the basis for forming error responses.

7.1.3 OSLC Servers **SHOULD** return an [Error resource](#) using the same representation format requested by the client via the HTTP **Accept** request header. [HTTP11]

7.1.4 OSLC Clients **SHOULD** treat the `oslc:statusCode` as a String that starts with digits, but may contain non-digit text.

### 7.2 Shape: Error

Used when servers may need a consistent shape to communicate error messages.

- **Name:** `core#Error`
- **URI:** `http://open-services.net/ns/core#Error`
- **Summary:** OSLC Core Error Shape

#### core#Error Properties

Prefixed Name	Occurs	Read-only	Value-type	Representation	Range	Description
<code>oslc:extendedError</code>	Zero-or-one	true	AnyResource	Either	<code>oslc:ExtendedError</code>	Extended error information
<code>oslc:message</code>	Exactly-one	true	string	N/A	Unspecified	An informative message describing the error that occurred.
<code>oslc:statusCode</code>	Exactly-one	true	string	N/A	Unspecified	The HTTP status code reported with the error.

### 7.3 Shape: ExtendedError

Additional details about an error the server had when processing the request.

- **Name:** `core#ExtendedError`
- **URI:** `http://open-services.net/ns/core#ExtendedError`
- **Summary:** OSLC Core ExtendedError Shape

#### core#ExtendedError Properties

Prefixed Name	Occurs	Read-only	Value-type	Representation	Range	Description
<code>oslc:hintHeight</code>	Zero-or-one	true	string	N/A	Unspecified	Values <b>MUST</b> be expressed in relative length units as defined in the W3C Cascading Style Sheets Specification (CSS 2.1) Em and ex units are interpreted relative to the default system font (at 100% size).
<code>oslc:hintWidth</code>	Zero-or-one	true	string	N/A	Unspecified	Values <b>MUST</b> be expressed in relative length units as defined in the W3C Cascading Style Sheets Specification (CSS 2.1) Em and ex units are interpreted relative to the default system font (at 100% size)
<code>oslc:moreInfo</code>	Zero-or-one	true	Resource	Reference	Unspecified	A resource giving more information on the error <b>SHOULD</b> be of an HTML content-type.
<code>oslc:rel</code>	Zero-or-one	true	string	N/A	Unspecified	If present and set to 'alternate' then indicates that work-around is provided, behavior for other values is undefined.

See [section 10. Terms](#) for details on error property terms.

## 8. Terms for describing vocabularies

## 8.1 Inverse Labels

The [W3C RDF Schema vocabulary](#) defines the vocabulary annotation property `rdfs:label`. This property is intended to provide a human-readable description for a resource's name. It is often used to provide a label for RDF properties. [\[LinkGuidance\]](#) discourages the creation of inverse predicates. However, there is still a need for a property, like `rdfs:label`, to specify an inverse label for a predicate.

For example, consider the OSLC Requirements Management (RM) property `oslc_rm:validatedBy`. When used as the predicate of a triple, this property is used to assert that the subject resource, e.g. a Requirement, is validated by the object resource, e.g. a TestCase. The `rdfs:label` for this property is "validatedBy".

Now consider the user interface of a query builder that allows users to build queries about TestCases. It is natural for the query builder to present the user with a list of the properties that apply to TestCases. Suppose the user wants to build a query that returns all the Requirements that are validated by a TestCase. The query builder should describe the available properties from the point of view of the TestCase. This implies that the query builder should describe the inverse relation asserted by any triple that has the TestCase as an object. In our example, the query builder should describe `oslc_rm:validatedBy` as "validates".

The `oslc:inverseLabel` property provides a human-readable label for the inverse of the subject property.

For example, the following triple (in Turtle notation) would be added to the OSLC RM vocabulary:

### EXAMPLE 3

```
oslc_rm:validatedBy oslc:inverseLabel "validates".
```

It should be noted that the use of inverse labels is independent of the existence of explicit RDF inverse properties. However, if an inverse property is defined by some vocabulary, then a consistent label should be used in order to avoid confusion. In general, it is good practice to avoid the creation of inverse properties since it creates redundant information and complicates SPARQL queries. Instead, a single property should be wherever possible and it should be given an inverse label in order to describe the property from the perspective of the object.

For example, the [OSLC Quality Management \(QM\) vocabulary](#) defines two properties that are approximately inverse to `oslc_rm:validatedBy`. These are `oslc_qm:validatesRequirement` and `oslc_qm:validatesRequirementCollection`. In this case the choice of inverse label "validates" for `oslc_rm:validatedBy` is consistent with the actual labels of the inverse properties, namely "validatesRequirement" and "validatesRequirementCollection".

## 8.2 Traceability and Impact type

Some RDF properties express dependency relations between artifacts, and it is often very valuable to trace the impact of a change in an artifact to those artifacts that depend on it directly or indirectly. The concept of dependency is very general. For example, the concept of trace relations is described in SysML: "A generic trace requirement relationship provides a general-purpose relationship between a requirement and any other model element. The semantics of trace include no real constraints and therefore are quite weak."

As a general guideline, if any assertion involving a given predicate may become invalid if the state of either its subject or object resources change, then we may legitimately regard that predicate as expressing a dependency relation, in which case it may be useful to explicitly describe the nature of the dependency.

An assertion describes a link between subject and object resources whose name is the property or predicate of the assertion. A dependency relationship may be in the same direction as the link, the opposite direction, both directions, or the link may not represent any dependency whose impact might need to be assessed.

For example, in assertions such as {requirement validatedBy testcase}, it may be important to assess the impact of a change in the requirement or a change in the testcase. Typically test cases are updated to reflect changes in requirements in order perform the correct validation. So in this case, property validatedBy would introduce impact that follows the link, from the subject requirement to the object testcase. However, if a team is doing test-driven development, they may treat test cases as formal, executable specifications of requirements and the requirement is simply an informal description of the test case. In this case, the team might consider the impact to be opposite of the link, from the testcase to the requirement.

The property `oslc:impactType` asserts that the subject property is a dependency relation and gives the direction of impact. The resources `oslc:FollowsLink` and `oslc:OppositeLink` identify whether the impact follows the direction of the assertion (subject to object), or the opposite direction (object to subject). `oslc:SymmetricImpact` describes a symmetric dependency relation in which the property represents a dependency from both subject to object and object to subject. `oslc:NoImpact` indicates the predicate does not represent any dependency between the subject and object resources.

For example, the following triple (in Turtle notation) would be added a vocabulary to indicate test cases are dependent on requirements:

### EXAMPLE 4

```
ex:validatedBy oslc:impactType oslc:FollowsLink .
```

The same dependency could also be described from the perspective of the test case. In this case, the dependency is opposite of the `validatesRequirement` predicate:

### EXAMPLE 5

```
ex:validatesRequirement oslc:impactType oslc:OppositeImpact .
ex:validatesRequirementCollection oslc:impactType oslc:OppositeImpact.
```

## 9. Discovery

Vocabulary terms are discovered via published vocabulary documents at the OSLC Core namespace and shapes at advertised URLs.

## 10. Terms

### 10.1 Vocabulary Details

The namespace URI for this vocabulary is: <http://open-services.net/ns/core#>

All vocabulary URIs defined in the OSLC Core namespace.

#### 10.1.1 RDFS Classes in this namespace

[AllowedValues](#), [AttachmentContainer](#), [AttachmentDescriptor](#), [Comment](#), [Compact](#), [CreationFactory](#), [Dialog](#), [Discussion](#), [Error](#), [ExtendedError](#), [OAuthConfiguration](#), [PrefixDefinition](#), [Preview](#), [Property](#), [Publisher](#), [QueryCapability](#), [ResourceShape](#), [ResponseInfo](#), [Service](#), [ServiceProvider](#), [ServiceProviderCatalog](#),

#### 10.1.2 RDF Properties in this namespace

[allowedValue](#), [allowedValues](#), [archived](#), [attachment](#), [attachmentSize](#), [authorizationURI](#), [comment](#), [creation](#), [creationDialog](#), [creationFactory](#), [default](#), [defaultValue](#), [describes](#), [details](#), [dialog](#), [discussedBy](#), [discussionAbout](#), [document](#), [domain](#), [executes](#), [extendedError](#), [futureAction](#), [hidden](#), [hintHeight](#), [hintWidth](#), [icon](#), [iconAltLabel](#), [iconSrcSet](#), [iconTitle](#), [impactType](#), [initialHeight](#), [inReplyTo](#), [instanceShape](#), [inverseLabel](#), [isMemberProperty](#), [label](#), [largePreview](#), [maxSize](#), [message](#), [modifiedBy](#), [moreInfo](#), [name](#), [nextPage](#), [oauthAccessTokenURI](#), [oauthConfiguration](#), [oauthRequestTokenURI](#), [occurs](#), [partOfDiscussion](#), [prefix](#), [prefixBase](#), [prefixDefinition](#), [property](#), [propertyDefinition](#), [queryBase](#), [queryCapability](#), [range](#), [readOnly](#), [rel](#), [representation](#), [resourceShape](#), [resourceType](#), [results](#), [selectionDialog](#), [service](#), [serviceProvider](#), [serviceProviderCatalog](#), [shortId](#), [shortTitle](#), [smallPreview](#), [statusCode](#), [totalCount](#), [usage](#), [valueShape](#), [valueType](#),

#### 10.1.3 RDF Descriptions(Individuals) in this namespace

[Any](#),

#### 10.1.4 AllowedValues

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

*AllowedValues* is an RDFS class.

Provides a way to specify allowed values for one or more properties.

#### 10.1.5 AttachmentContainer

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

*AttachmentContainer* is an RDFS class.

An LDP-C that contains attachments for a resource.

#### 10.1.6 AttachmentDescriptor

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

*AttachmentDescriptor* is an RDFS class.

An LDP-RS that contains additional data about an attachment.

#### 10.1.7 Comment

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

*Comment* is an RDFS class.

A Comment resource represents a single note, or comment, in a discussion thread.

#### 10.1.8 Compact

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

*Compact* is an RDFS class.

A resource describing how to display a link and Preview for another, associated resource.

#### **10.1.9 CreationFactory**

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

*CreationFactory* is an RDFS class.

The CreationFactory definition included in a ServiceProvider.

#### **10.1.10 Dialog**

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

*Dialog* is an RDFS class.

Describes information about a dialog such as its title and dimensions.

#### **10.1.11 Discussion**

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

*Discussion* is an RDFS class.

A Discussion resource is intended to represent a sequence of comments or notes regarding the associated resource

#### **10.1.12 Error**

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

*Error* is an RDFS class.

Basis for forming an error response.

#### **10.1.13 ExtendedError**

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

*ExtendedError* is an RDFS class.

Extended error information.

#### **10.1.14 OAuthConfiguration**

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

*OAuthConfiguration* is an RDFS class.

The OAuthConfiguration definition included in ServiceProvider.

#### **10.1.15 PrefixDefinition**

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

*PrefixDefinition* is an RDFS class.

The PrefixDefinition definition included in ServiceProvider.

#### **10.1.16 Preview**

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

*Preview* is an RDFS class.

An HTML representation of a resource that can be embedded in another user interface.

#### **10.1.17 Property**

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

*Property* is an RDFS class.

A Property resource describes one allowed or required property of a resource.

#### **10.1.18 Publisher**

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

*Publisher* is an RDFS class.

The *Publisher* definition included in *ServiceProvider*.

#### 10.1.19 QueryCapability

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

*QueryCapability* is an RDFS class.

The *QueryCapability* definition included in a *ServiceProvider*.

#### 10.1.20 ResourceShape

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

*ResourceShape* is an RDFS class.

The Resource Shape used for creation, query and modify. Formally, a shape *S* applies to a resource *R* if there is a triple *R* *rdf:type* *T* and there is a triple *S* *oslc:describes* *T*, or if there is a triple *R* *oslc:instanceShape* *S*.

#### 10.1.21 ResponseInfo

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

*ResponseInfo* is an RDFS class.

The *ResponseInfo* included in query results.

#### 10.1.22 Service

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

*Service* is an RDFS class.

The *Service* definition included in a *ServiceProvider*.

#### 10.1.23 ServiceProvider

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

*ServiceProvider* is an RDFS class.

The *Service Provider* resource

#### 10.1.24 ServiceProviderCatalog

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

*ServiceProviderCatalog* is an RDFS class.

The *Service Provider Catalog* resource

#### 10.1.25 allowedValue

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

*allowedValue* is an RDF property.

Specifies the allowed values for a property (may be more than one).

#### 10.1.26 allowedValues

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

*allowedValues* is an RDF property.

Reference to an *AllowedValues* resource that specifies the allowed values for the property.

#### 10.1.27 archived

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

*archived* is an RDF property.

Indicates whether the subject has been marked as archived, no longer an actively updating resource.

### 10.1.28 attachment

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

*attachment* is an RDF property.

An attachment associated with a resource. May be used as a membership predicate for an attachment container.

### 10.1.29 attachmentSize

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

*attachmentSize* is an RDF property.

Size in bytes of the attachment content.

### 10.1.30 authorizationURI

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

*authorizationURI* is an RDF property.

URI for obtaining OAuth authorization.

### 10.1.31 comment

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

*comment* is an RDF property.

Comment about the resource.

### 10.1.32 creation

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

*creation* is an RDF property.

To create a new resource via the factory, post it to this URI.

### 10.1.33 creationDialog

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

*creationDialog* is an RDF property.

Enables clients to create a resource via UI.

### 10.1.34 creationFactory

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

*creationFactory* is an RDF property.

Enables clients to create new resources.

### 10.1.35 default

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

*default* is an RDF property.

Used in conjunction with `oslc:usage` property used to identify which service is the default usage.

### 10.1.36 defaultValue

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

*defaultValue* is an RDF property.

A default value for property, inlined into property definition.

### 10.1.37 describes

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

*describes* is an RDF property.

This shape describes resources that are of the RDF type given by the object of the `oslc:describes` predicate. Formally, a shape `S` applies to a resource `R` if there is a triple `R rdf:type T` and there is a triple `S oslc:describes T`.

#### 10.1.38 details

`http://open-services.net/ns/core#details`

*details* is an RDF property.

A URL that may be used to retrieve a resource to determine additional details about the service provider.

#### 10.1.39 dialog

`http://open-services.net/ns/core#dialog`

*dialog* is an RDF property.

The URI of the HTML dialog.

#### 10.1.40 discussedBy

`http://open-services.net/ns/core#discussedBy`

*discussedBy* is an RDF property.

A series of notes and comments about this resource.

#### 10.1.41 discussionAbout

`http://open-services.net/ns/core#discussionAbout`

*discussionAbout* is an RDF property.

Reference to associated resource.

#### 10.1.42 document

`http://open-services.net/ns/core#document`

*document* is an RDF property.

The URI of an HTML document to be used for the preview.

#### 10.1.43 domain

`http://open-services.net/ns/core#domain`

*domain* is an RDF property.

Namespace URI of the specification that is implemented by this service. In most cases this namespace URI will be for an OSLC domain, but other URIs **MAY** be used.

#### 10.1.44 executes

`http://open-services.net/ns/core#executes`

*executes* is an RDF property.

Link from a currently available action to the future action it realizes.

#### 10.1.45 extendedError

`http://open-services.net/ns/core#extendedError`

*extendedError* is an RDF property.

Extended (additional) error information.

#### 10.1.46 futureAction

`http://open-services.net/ns/core#futureAction`

*futureAction* is an RDF property.

A predicate that links to an action that is not currently executable on the subject resource, but may be executable in the future and/or on other resources. For example, in OSLC Automation this is expected to link from an `oslc_auto:AutomationPlan` to an `oslc:Action` resource with zero bindings (as it is not executable), with the meaning that the executable form of the action may be available on

oslc\_auto:AutomationResult resources generated by executing that Automation Plan. Similarly, resource shapes can allow discovery of actions available on the output of a creation factory.

#### 10.1.47 hidden

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

*hidden* is an RDF property.

A hint that indicates that property **MAY** be hidden when presented in a user interface.

#### 10.1.48 hintHeight

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

*hintHeight* is an RDF property.

Preferred height of a delegated user interface. Values must be expressed using length units as specified in Cascading Style Sheets 2.1.

#### 10.1.49 hintWidth

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

*hintWidth* is an RDF property.

Preferred width of a delegated user interface. Values must be expressed using length units as specified in Cascading Style Sheets 2.1.

#### 10.1.50 icon

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

*icon* is an RDF property.

URI of an image applicable to the resource.

#### 10.1.51 iconAltLabel

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

*iconAltLabel* is an RDF property.

Alternative label used in association with the `oslc:icon`, such as HTML `img` tag's `alt` attribute.

#### 10.1.52 iconSrcSet

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

*iconSrcSet* is an RDF property.

Specification of a set of images of different sizes based on HTML `img` element `srcset` attribute.

#### 10.1.53 iconTitle

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

*iconTitle* is an RDF property.

Title used in association with the `oslc:icon`, such as HTML `img` tag's `title` attribute.

#### 10.1.54 impactType

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

*impactType* is an RDF property.

Asserts that the subject property is a dependency relation and gives the direction of impact.

#### 10.1.55 initialHeight

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

*initialHeight* is an RDF property.

Recommended initial height of the preview. The presence of this property indicates that the preview supports dynamically computing its size. Values **MUST** be expressed in relative length units as defined in the W3C Cascading Style Sheets Specification (CSS 2.1). `em` and `ex` units are interpreted relative to the default system font (at 100% size).

#### 10.1.56 inReplyTo

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

*inReplyTo* is an RDF property.

Reference to comment this comment is in reply to.

#### 10.1.57 **instanceShape**

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

*instanceShape* is an RDF property.

The URI of a Resource Shape that describes the possible properties.

#### 10.1.58 **inverseLabel**

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

*inverseLabel* is an RDF property.

Provides a human-readable label for the inverse of the subject property.

#### 10.1.59 **isMemberProperty**

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

*isMemberProperty* is an RDF property.

Used to define when a property is a member of a container, useful for query.

#### 10.1.60 **label**

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

*label* is an RDF property.

Very short label for use in menu items.

#### 10.1.61 **largePreview**

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

*largePreview* is an RDF property.

URI and sizing properties for an HTML document to be used for a large preview.

#### 10.1.62 **maxSize**

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

*maxSize* is an RDF property.

For String properties only, specifies maximum characters allowed. If not set, then there is no maximum or maximum is specified elsewhere.

#### 10.1.63 **message**

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

*message* is an RDF property.

An informative message describing the error that occurred.

#### 10.1.64 **modifiedBy**

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

*modifiedBy* is an RDF property.

The URI of a resource describing the entity that most recently modified this resource. The link target is usually a foaf:Person or foaf:Agent, but could be any type. This is modeled after dcterms:creator, but Dublin Core currently has no equivalent property.

#### 10.1.65 **moreInfo**

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

*moreInfo* is an RDF property.

A resource giving more information on the error **SHOULD** be of an HTML content-type.

#### 10.1.66 name

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

*name* is an RDF property.

Name of property being defined, i.e. second part of property's Prefixed Name. For all other uses, consider `dcterms:title`, `rdfs:label`, `oslc:shortTitle` or `oslc:label`

#### 10.1.67 nextPage

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

*nextPage* is an RDF property.

Link to next page of response.

#### 10.1.68 oauthAccessTokenURI

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

*oauthAccessTokenURI* is an RDF property.

URI for obtaining OAuth access token.

#### 10.1.69 oauthConfiguration

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

*oauthConfiguration* is an RDF property.

Defines the three OAuth URIs required for a client to act as an OAuth consumer.

#### 10.1.70 oauthRequestTokenURI

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

*oauthRequestTokenURI* is an RDF property.

URI for obtaining OAuth request token.

#### 10.1.71 occurs

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

*occurs* is an RDF property.

**MUST** be either <http://open-services.net/ns/core#Exactly-one>, <http://open-services.net/ns/core#Zero-or-one>, <http://open-services.net/ns/core#Zero-or-many> or <http://open-services.net/ns/core#One-or-many>.

#### 10.1.72 partOfDiscussion

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

*partOfDiscussion* is an RDF property.

Reference to owning Discussion resource .

#### 10.1.73 prefix

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

*prefix* is an RDF property.

Namespace prefix to be used for this namespace.

#### 10.1.74 prefixBase

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

*prefixBase* is an RDF property.

The base URI of the namespace.

#### 10.1.75 prefixDefinition

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

*prefixDefinition* is an RDF property.

Defines a namespace prefix for use in JSON representations and in forming OSLC Query Syntax strings.

#### 10.1.76 property

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

*property* is an RDF property.

The properties that are allowed or required by this shape.

#### 10.1.77 propertyDefinition

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

*propertyDefinition* is an RDF property.

URI of the property whose usage is being described.

#### 10.1.78 queryBase

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

*queryBase* is an RDF property.

The base URI to use for queries. Queries may be invoked either by HTTP GET or HTTP POST. For HTTP GET, a query URI is formed by appending a key=value pair to the base URI. For HTTP POST, the query parameters are encoded as content with media type application/x-www-form-urlencoded and sent in the request body. The base URI **MAY** accept other query languages and media types in the request body, e.g. application/sparql-query for SPARQL queries.

#### 10.1.79 queryCapability

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

*queryCapability* is an RDF property.

Enables clients query across a collection of resources.

#### 10.1.80 range

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

*range* is an RDF property.

For properties with a resource value-type, Providers **MAY** also specify the range of possible resource types allowed, each specified by URI. The default range is <http://open-services.net/ns/core#Any>.

#### 10.1.81 readOnly

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

*readOnly* is an RDF property.

true if the property is read-only. If omitted, or set to false, then the property is writable. Providers **SHOULD** declare a property read-only when changes to the value of that property will not be accepted after the resource has been created, e.g. on PUT/PATCH requests. Consumers should note that the converse does not apply: Providers **MAY** reject a change to the value of a writable property.

#### 10.1.82 rel

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

*rel* is an RDF property.

If present and set to 'alternate' then indicates that work-around is provided, behavior for other values is undefined.

#### 10.1.83 representation

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

*representation* is an RDF property.

Should be <http://open-services.net/ns/core#Reference>, <http://open-services.net/ns/core#Inline> or <http://open-services.net/ns/core#Either>

#### 10.1.84 resourceShape

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

*resourceShape* is an RDF property.

A Creation Factory **MAY** provide Resource Shapes that describe shapes of resources that may be created.

#### 10.1.85 resourceType

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

*resourceType* is an RDF property.

The expected resource type URI of the resource that will be created using this creation factory. These would be the URIs found in the result resource's `rdf:type` property.

#### 10.1.86 results

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

*results* is an RDF property.

Used to hold the results of dialog action or JSON query results (default). The JSON query result attribute 'oslc:results' is used whenever a provider doesn't have a suitable property already in its model for such purposes.

#### 10.1.87 selectionDialog

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

*selectionDialog* is an RDF property.

Enables clients to select a resource via a UI.

#### 10.1.88 service

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

*service* is an RDF property.

Describes a service offered by the service provider.

#### 10.1.89 serviceProvider

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

*serviceProvider* is an RDF property.

A link to the resource's OSLC Service Provider.

#### 10.1.90 serviceProviderCatalog

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

*serviceProviderCatalog* is an RDF property.

Additional service provider catalog.

#### 10.1.91 shortId

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

*shortId* is an RDF property.

A short, human-readable, plain text value. This value should be unique in some context that is apparent to human users of a service.

#### 10.1.92 shortTitle

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

*shortTitle* is an RDF property.

Shorter form of `dcterms:title` for the resource.

#### 10.1.93 smallPreview

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

*smallPreview* is an RDF property.

URI and sizing properties for an HTML document to be used for a small preview.

#### 10.1.94 statusCode

`http://open-services.net/ns/core#statusCode`

*statusCode* is an RDF property.

The HTTP status code reported with the error.

#### 10.1.95 totalCount

`http://open-services.net/ns/core#totalCount`

*totalCount* is an RDF property.

This optional property indicates the total number of results across all pages, its value should be non-negative. In the context of a query resource, this value **SHOULD** be the total number of results, i.e. the number of resources that match the query. In the context of other resources, the value **SHOULD** be the total number of property values (i.e. RDF triples) of the resource. Unless Stable Paging is in effect, the total count **MAY** vary as a client retrieves subsequent pages.

#### 10.1.96 usage

`http://open-services.net/ns/core#usage`

*usage* is an RDF property.

An identifier URI for the domain specified usage of this creation factory. If a service provides multiple creation factories, it may designate the primary or default one that should be used with a property value of `http://open-services.net/ns/core#default`.

#### 10.1.97 valueShape

`http://open-services.net/ns/core#valueShape`

*valueShape* is an RDF property.

if the value-type is a resource type, then Property **MAY** provide a shape value to indicate the Resource Shape that applies to the resource.

#### 10.1.98 valueType

`http://open-services.net/ns/core#valueType`

*valueType* is an RDF property.

A URI that indicates the value type, for example XML Schema or RDF URIs for literal value types, and OSLC-specified for others. If this property is omitted, then the value type is unconstrained.

#### 10.1.99 Any

`http://open-services.net/ns/core#Any`

*Any* is an RDF description.

Any value type is allowed.

## Appendix A. Change History

*This section is non-normative.*

Revision	Date	Editor	Changes Made
1	07/06/2016	Jim Amsden	CSD was approved and published.