OSLC Availability Specification Draft 0.6

Status: 0.6 Draft Specification - 1 July 2014

This Version

- OSLC Availability Specification Version 0.5

Latest Version

- OSLC Availability Specification Version 0.5

PreviousVersion

- This specification is the initial version of an OSLC Availability specification.

Authors

- Tim Friessinger (IBM)
- Jürgen Holtz (IBM)

Table of Contents

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 builds on the OSLC Core Specification to define the resources and operations

supported by an Open Services for Lifecycle Collaboration (OSLC) Availability provider.

The Availability Specifications is intended to represent the availability of a system's resources across their life cycle with particular focus on high availability. With a system we mean in general a software system, but other types of systems are explicitly not excluded.

To be able to describe a system that is (highly) available, it is important to understand the state (the “health”) of the system's resources.

An implementation of this specification allows a consumer to see what resources belong to a system at all, the state/ health of them (are they currently available/ online, unavailable/ offline or in a problem state etc.) and how they are organized to ensure high availability.

The intent of this specification is to define the set of HTTP-based RESTful interfaces in terms of HTTP methods: GET, POST, PUT and DELETE, HTTP response codes, mime type handling and resource formats. The capabilities of the interface definitions are driven by key integration scenarios and therefore don’t represent a complete setup of operations on resources or resource types. The resource formats and operations may not match exactly the native models supported by availability service providers but are intended to be compatible with them.

Terminology

Service Provider - an implementation of the OSLC Availability specification as a server. OSLC Availability clients consume these services.

Availability Resource – Defines a resource of the (software) system, that is described by the Availability service provider. Availability Resources are stateful and can change their appearance during their lifetime – for example as result of a process executed externally from this specification.

Availability Condition - Defines the condition of an Availability Resource, including its current and desired state, measured MTTR values etc.

Availability Group - Several Availability Resources, grouped under the context of (high) availability. For example a group of redundant services, that can handle a fail-over scenario.

Redundancy Group – Group of Availability Resources that are redundant to each other. Recommended as a specialization of an Availability Group.

Redundancy Member – Member of a Redundancy Group. Recommended as a specialization of an Availability Resource.

Base Requirements

Compliance

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

The following table summarizes the requirements from the OSLC Core Specification as well as some (but not all) additional requirements specific to Availability. See the full content of the

Availability specification for all requirements. Note that this specification further restricts some of the requirements for OSLC Core Specification as noted in the Origin column of the compliance table. See further sections in this specification or the OSLC Core Specification to get further details on each of these requirements.

Any consumer or service provider behaviours are allowed unless explicitly prohibited by this or dependent specifications; conditional permissive requirements, especially those qualified with “MAY”, are implicitly covered by the preceding clause. While technically redundant in light of that broad permission, OSLC specifications do still make explicit MAY-qualified statements in cases where the editors believe doing so is likely to add clarity.

Requirements on OSLC Consumers

Requirement	*Level*	*Origin(s)*	*Meaning*
Unknown properties and content	MUST	Core	OSLC clients MUST preserve unknown content
Unknown properties and content	SHOULD	Core	OSLC clients SHOULD assume an OSLC service will discard unknown property values.

Requirements on OSLC Service Providers

Requirement	*Level*	*Origin(s)*	*Meaning*
Unknown properties and content	MUST	Core	OSLC service providers MUST return an error code if recognized content is invalid.
Unknown properties and content	SHOULD	Core	OSLC service providers SHOULD NOT return an error code for unrecognized content.
Unknown properties and content	MAY	Core	OSLC service providers MAY ignore unknown content
Resource Operations	MUST	Core	OSLC service providers MUST support resource operations via standard HTTP operations
Resource Paging	MAY	Core	OSLC services MAY provide paging for resources
Partial Resource Representations	SHOULD	Core	OSLC service providers SHOULD support HTTP GET requests for retrieval of a subset of a resource’s properties via the oslc.properties URL parameter
Partial Resource Representations	MAY	Core	OSLC service providers MAY support HTTP PUT requests for updating a subset of a resource’s properties via the oslc.properties URL parameter
Service Provider Resources	MAY	Core	OSLC service providers MAY provide a Service Provider Catalog resource
Service Provider Resources	MUST	Core	OSLC service providers MUST provide a Service Provider resource
Creation Factories	MAY	Core	OSLC service providers MAY provide creation factories to enable resource creation via HTTP POST
Query Capabilities	SHOULD [1]	Availability, Core	OSLC service providers SHOULD provide query capabilities to enable clients to query for resources
Query Syntax	MUST [2]	Availability, Core	If a service provider supports OSLC query capabilities, the query capabilities MUST support the OSLC Core Query Syntax

Requirement	*Level*	*Origin(s)*	*Meaning*
Query Syntax	MAY	Core	OSLC query capabilities MAY support other query syntax
Delegated UI Dialogs	SHOULD	Core	OSLC service providers SHOULD allow clients to discover, via their service provider resources, any Delegated UI Dialogs they offer.
Delegated UI Dialogs	SHOULD	Core	OSLC service providers SHOULD offer delegated UI dialogs for resource creation
Delegated UI Dialogs	SHOULD	Core	OSLC service providers SHOULD offer delegated UI dialogs for resource selection
UI Preview	SHOULD	Core	OSLC Services SHOULD offer UI previews for resources that may be referenced by other resources
HTTP Basic Authentication	MAY	Core	OSLC Services MAY support Basic Auth
HTTP Basic Authentication	SHOULD	Core	OSLC Services SHOULD support Basic Auth only over HTTPS
OAuth Authentication	MAY	Core	OSLC service providers MAY support OAuth
OAuth Authentication	SHOULD	Core	OSLC service providers that support OAuth SHOULD allow clients to discover the required OAuth URLs via their service provider resource
Error Responses	MAY	Core	OSLC service providers MAY provide error responses using Core-defined error formats
RDF/XML Representations	MUST[3]	Availability, Core	OSLC service providers MUST offer an RDF/XML representation for HTTP GET responses
RDF/XML Representations	MUST[3]	Availability, Core	OSLC service providers MUST accept RDF/XML representations on PUT requests.
RDF/XML Representations	MUST[3]	Availability, Core	OSLC service providers MUST accept RDF/XML representations on POST requests whose semantic intent is to create a new resource instance.
XML Representations	MAY[3]	Availability, Core	OSLC service providers MAY provide a XML representation for HTTP GET, POST and PUT requests that conform to the Core Guidelines for XML.
JSON Representations	MAY[3]	Availability, Core	OSLC service providers MAY provide JSON representations for HTTP GET, POST and PUT requests that conform to the Core Guidelines for JSON
HTML Representations	SHOULD	Availability, Core	OSLC service providers SHOULD provide HTML representations for HTTP GET requests

- [1]The OSLC Core Specifications indicates service providers MAY provide Query Capabilities. This specification for OSLC Availability makes Query Capability support a SHOULD requirement.
- [2]The OSLC Core Specifications indicates service providers MAY support the OSLC Query Syntax. This specification for OSLC Availability makes OSLC Query Syntax support a MUST requirement for service providers providing query capabilities.

- [3]Support for all HTTP methods for all availability resources is not required. See the HTTP Method support table for details.

Specification Versioning

See OSLC Core Specification Versioning section.

Namespaces

Defined

OSLC Availability defines the namespace shown in the table below. This namespace URI and prefix are used to designate the resources and their properties defined in this specification.

Use of the suggested prefix is RECOMMENDED, because doing so aids debugging and other situations where humans read the data.

Suggested namespace prefix	Namespace URI
`oslc-availability`	`http://open-services.net/ns/availability#`

Re-used from other specifications

In addition to the namespace URIs and namespace prefixes defined in the OSLC Core specification, OSLC Avail also re-uses vocabulary terms from other namespaces. The namespace prefixes in the table below are used in this specification, and match the recommendations made by the specification that defines each.

Namespace prefix used	Namespace URI	Usage
`crtv`	`http://open-services.net/ns/crtv#`	Vocabulary is expected to be commonly used by Availability providers, but is not required. Defined in the OSLC Reconciliation domain.
`ems`	`http://open-services.net/ns/ems#`	Vocabulary is required for Availability providers to expose metrics. Defined in the OSLC Estimation and Measurement domain.
`oslc_asset`	`http://open-services.net/ns/asset#`	Vocabulary is expected to be commonly used by Availability providers, but is not required. Defined in the OSLC Asset Management domain.
`oslc_auto`	`http://open-services.net/ns/auto#`	Vocabulary is expected to be commonly used by Availability providers, but is not required. Defined in the OSLC Automation domain.
`oslc_rm`	`http://open-services.net/ns/rm#`	Vocabulary is expected to be commonly used by Availability providers, but is not required. Defined in the OSLC Requirements Management domain.

Resource Formats

In addition to the requirements for OSLC Defined Resource Representations, this section outlines further refinements and restrictions.

See HTTP Method support table for further clarification on support for HTTP methods and media types for each OSLC Availability resource.

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

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

For HTTP PUT/POST request formats for Availability resources,

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

For HTTP GET response formats for Query requests,

Availability Providers **MUST** provide RDF/XML and **MAY** provide JSON, XML, and Atom Syndication Format XML.

When Availability Consumers request:

- `application/rdf+xml` Availability Providers **MUST** respond with RDF/XML representation without restrictions.
- `application/xml` Availability Providers **SHOULD** respond with OSLC-defined abbreviated XML representation as defined in the OSLC Core Representations Guidance
- `application/atom+xml` Availability Providers **SHOULD** respond with Atom Syndication Format XML representation as defined in the OSLC Core Representations Guidance
- If supported, the Atom Syndication Format XML representation **SHOULD** use RDF/XML representation without restrictions for the atom:content entries representing the resource representations.

Authentication

See OSLC Core Authentication section. OSLC Availability puts no additional constraints on authentication.

Error Responses

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

Pagination

OSLC Availability 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.

Labels for Relationships

Availability relationships to other resources are represented as properties whose values are the URI of the object or target resource. When an Availability relationship property is to be presented in a user interface, it may be helpful to provide an informative and useful textual label for that relationship instance. (This in addition to the relationship property URI and the object resource URI, which are also candidates for presentation to a user.) To this end, OSLC providers **MAY** support a `dcterms:title` link property in Availability resource representations, using the anchor approach outlined in the OSLC Core Links Guidance.

RDF/XML and XML example using reified statement:

```
<rdf:RDF
     xmlns:dcterms="http://purl.org/dc/terms/"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:oslc_rm="http://open-services.net/ns/rm#"
     xmlns:oslc-availability="http://open-services.net/ns/availability#">

    <oslc-availability:AvailabilityResource
rdf:about="http://example.com/components/1234">
      <dcterms:title>Pet Shop App production webserver</dcterms:title>
      <oslc_rm:affectedBy rdf:ID="link1"
             rdf:resource="http://example.com/plans/123" />
    </oslc_auto:AvailabilityResource>

    <rdf:Description rdf:about="#link1">
      <dcterms:title>Automation Plan 123: Pet Shop App production
build</dcterms:title>
   </rdf:Description>
</rdf:RDF>
```

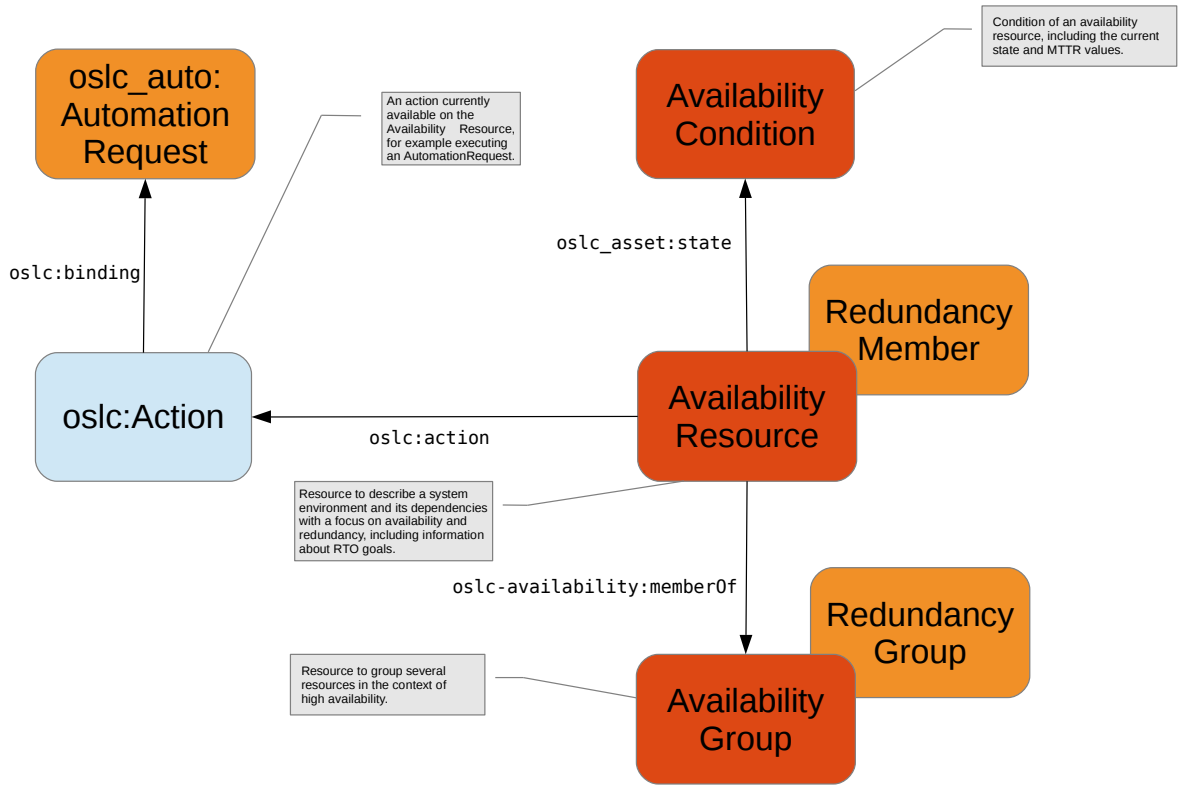
Availability Definitions

The Availability properties are not limited to the ones defined in this specification; service providers may provide additional properties. Any additional properties SHOULD exist in their own unique namespace and not use the namespaces defined in this specification.

A list of properties is defined for each type of resource. Most of these properties are identified in

OSLC Core Appendix A: Common Properties. Any exceptions are noted. Relationship properties refer to other resources. These resources may be in any OSLC domain (including Availability).

The diagram below shows the relationships between the Availability specification's resources.



For all resource types defined in this specification, all **required** properties (those defined with an occurrence of **exactly-one** or **one-or-many**) **MUST** exist for each resource and must be provided when requested. All other properties are optional, and might not exist on some or any resources; those that do not exist will not be present in the returned representation even if requested, while those that do exist **MUST** be provided if requested. Providers **MAY** define additional provider-specific properties; providers **SHOULD** use their own namespaces for such properties, or use standard Dublin Core or RDF namespaces and properties where appropriate.

If no specific set of properties is requested, **all** properties are returned - both those defined in this specification as well as any provider-specific ones. See Selective Property Values in OSLC Core Specification.

Resource: AvailabilityResource

- **Name:** `AvailabilityResource`
- **Description:** A resource in a (complex) system environment in the context of (high) availability.
- **Type URI** `http://open-services.net/ns/availability#AvailabilityResource`

AvailabilityComponent Properties

<table>
<tr><th>Prefixed Name</th><th>Details</th></tr>
<tr><td colspan="2">OSLC Core: Common Properties</td></tr>
<tr><td>oslc:action</td><td>
<table>
<tr><td>Occurs</td><td>zero-or-many</td></tr>
<tr><td>Read-only</td><td>unspecified</td></tr>
<tr><td>Value-type</td><td>Resource</td></tr>
<tr><td>Representation</td><td>Either</td></tr>
<tr><td>Range</td><td>any</td></tr>
<tr><td>Description</td><td>An action currently available on the subject resource, i.e. it links to an action that the provider asserts is currently available for execution by clients at the time the representation was formed.

A typical scenario for an action on an AvailabilityResource is to change its condition, e.g. to update its oslc-availability:desiredState so that a resource, representing a stopped software system, will be started.

In a distributed system, clients can lose race conditions that result in an “available”-appearing action’s execution requests being rejected. It is likely that the target resource will be an oslc-availability:ChangeConditionAction but that is not necessarily the case; when it is an <code>oslc-availability:ConditionAction</code>, it will have at least one binding since it is currently available.</td></tr>
</table>
</td></tr>
<tr><td>dcterms:created</td><td>
<table>
<tr><td>Occurs</td><td>zero-or-one</td></tr>
<tr><td>Read-only</td><td>True</td></tr>
<tr><td>Value-type</td><td>DateTime</td></tr>
<tr><td>Representation</td><td>Inline</td></tr>
<tr><td>Range</td><td>any</td></tr>
<tr><td>Description</td><td>Timestamp of resource creation (reference: Dublin Core)</td></tr>
</table>
</td></tr>
<tr><td>dcterms:creator</td><td>
<table>
<tr><td>Occurs</td><td>zero-or-many</td></tr>
<tr><td>Read-only</td><td>unspecified</td></tr>
<tr><td>Value-type</td><td>AnyResource</td></tr>
<tr><td>Representation</td><td>Either</td></tr>
<tr><td>Range</td><td>any</td></tr>
<tr><td>Description</td><td>Creator or creators of resource (reference: Dublin Core). It is likely that the target</td></tr>
</table>
</td></tr>
</table>

	resource will be an `foaf:Person` but that is not necessarily the case. For example the execution of an `oslc_auto:AutomationPlan` can also be responsible for the creation of an availability resource
`dcterms:description`	**Occurs**: zero-or-one **Read-only**: unspecified **Value-type**: XMLLiteral **Representation**: Inline **Range**: any **Description**: Descriptive text (reference: Dublin Core) about the resource represented as rich text in XHTML content. SHOULD include only content that is valid and suitable inside an XHTML element.
`dcterms:identifier`	**Occurs**: exactly-one **Read-only**: True **Value-type**: String **Representation**: Inline **Range**: any **Description**: A unique identifier for a resource. Assigned by the service provider when a resource is created. Not intended for end-user display.
`oslc:instanceShape`	**Occurs**: zero-or-one **Read-only**: True **Value-type**: Resource **Representation**: Reference **Range**: `oslc:ResourceShape` **Description**: Resource Shape that provides hints as to resource property value-types and allowed values.
`dcterms:modified`	**Occurs**: zero-or-one **Read-only**: True **Value-type**: DateTime **Representation**: Inline **Range**: any **Description**: Timestamp of latest resource modification (reference: Dublin Core). **Note:** An modification to an Availability Resource's AvailabilityCondition

	(oslc_asset:state) is not a resource modification, and therefore is not reflected by an update of this property. To see the currentness of an AvailabilityCondition, refer to it's dcterms:created field.

`oslc:serviceProvider`

Occurs	zero-or-many
Read-only	True
Value-type	Resource
Representation	Reference
Range	`oslc:ServiceProvider`
Description	The scope of a resource is a link to the resource's OSLC Service Provider.

`oslc_asset:state`

Occurs	exactly-one
Read-only	unspecified
Value-type	AnyResource
Representation	Either
Range	`oslc-availability:AvailabilityCondition`
Description	Detailed information about this the component's state. Either a local (inline) or referenced resource and use the attributes (the range) of the `oslc-availability:AvailabilityCondition` resource.

`dcterms:subject`

Occurs	zero-or-many
Read-only	unspecified
Value-type	String
Representation	Inline
Range	any
Description	Tag or keyword for a resource. Each occurrence of a dc:subject property denotes an additional tag for the resource.

`dcterms:title`

Occurs	exactly-one
Read-only	unspecified
Value-type	XMLLiteral
Representation	Inline
Range	any
Description	A name given to the resource (reference: Dublin Core). If unique, it can be used to identify different automation resources.

<table>
<tr><td>rdf:type</td><td><table>
<tr><td>Occurs</td><td>zero-or-many</td></tr>
<tr><td>Read-only</td><td>unspecified</td></tr>
<tr><td>Value-type</td><td>Resource</td></tr>
<tr><td>Representation</td><td>Reference</td></tr>
<tr><td>Range</td><td>any</td></tr>
<tr><td>Description</td><td>The resource type URIs.</td></tr>
</table></td></tr>
<tr><td colspan="2">OSLC AvailabilityResource: Start of additional properties including relationship properties.</td></tr>
<tr><td>oslc-availability:memberOf</td><td><table>
<tr><td>Occurs</td><td>zero-or-many</td></tr>
<tr><td>Read-only</td><td>unspecified</td></tr>
<tr><td>Value-type</td><td>Resource</td></tr>
<tr><td>Representation</td><td>Reference</td></tr>
<tr><td>Range</td><td>any</td></tr>
<tr><td>Description</td><td>AvailabilityGroup, this AvailabilityResource is a member of and therefore is in a special high availability relationship with the other members. It is expected that the target of this link will be of type Availability Group (and/or Redundancy Group), but this is not necessarily the case.</td></tr>
</table></td></tr>
<tr><td>oslc-availability:rto</td><td><table>
<tr><td>Occurs</td><td>zero-or-one</td></tr>
<tr><td>Read-only</td><td>unspecified</td></tr>
<tr><td>Value-type</td><td>AnyResource</td></tr>
<tr><td>Representation</td><td>Either</td></tr>
<tr><td>Range</td><td>any</td></tr>
<tr><td>Description</td><td>Recovery Time Objective (RTO) for this AvailabilityResource. The maximum allowable period of time, the resource can be offline or in a problematic state.
It is likely that the target resource will be an ems:Measure but that is not necessarily the case.</td></tr>
</table></td></tr>
</table>

Resource: AvailabilityCondition

- **Name:** AvailabilityCondition
- **Description:** A resource representing the current condition of an *AvailabilityResource*.
- **Type URI** http://open-services.net/ns/availability#AvailabilityCondition

AvailabilityCondition Properties

Prefixed Name	*Details*

OSLC Core: Common Properties	
`dcterms:contributor`	

Occurs	zero-or-many
Read-only	unspecified
Value-type	AnyResource
Representation	Either
Range	any
Description	Contributor or contributors who is/ are responsible for the current condition/ state of an Availability Component (reference: Dublin Core). It is likely that the target resource will be an `foaf:Person` but that is not necessarily the case. It can be for example also another `oslc-availability:AvailabilityResource` or an `auto:AutomationPlan.`

`dcterms:created`

Occurs	exactly-one
Read-only	True
Value-type	DateTime
Representation	Inline
Range	any
Description	Timestamp of resource creation (reference: Dublin Core): The point in time this condition has occurred. (“Snapshot of states.”)

`dcterms:description`

Occurs	zero-or-one
Read-only	unspecified
Value-type	XMLLiteral
Representation	Inline
Range	any
Description	Descriptive text (reference: Dublin Core) about the resource represented as rich text in XHTML content. SHOULD include only content that is valid and suitable inside an XHTML element. Can be used as a log to explain why a Availability Component is in this actual condition.

`dcterms:identifier`

Occurs	exactly-one
Read-only	True
Value-type	String
Representation	Inline
Range	any
Description	A unique identifier for a resource. Assigned

	by the service provider when a resource is created. Not intended for end-user display.

dcterms:title	
Occurs	exactly-one
Read-only	unspecified
Value-type	XMLLiteral
Representation	Inline
Range	any
Description	A name given to the resource (reference: Dublin Core). If unique, it can be used to identify different automation resources.

oslc:instanceShape	
Occurs	zero-or-one
Read-only	True
Value-type	Resource
Representation	Reference
Range	oslc:ResourceShape
Description	Resource Shape that provides hints as to resource property value-types and allowed values.

rdf:type	
Occurs	zero-or-many
Read-only	unspecified
Value-type	Resource
Representation	Reference
Range	any
Description	The resource type URIs.

OSLC AvailabilityCondition: Start of additional properties including relationship properties.

oslc-availability:compoundState	
Occurs	zero-or-one
Read-only	unspecified
Value-type	AnyResource
Representation	Either
Range	any
Description	Used to summarize the relationship between oslc-availability:currentState and oslc-availability:desiredState. This property can be used to see at the first glance if there is a mismatch between the desired and the detected (current) state. It is expected that this will be a resource reference to a definition of a valid target type on the service provider.

`oslc-availability:currentState`	**Occurs** exactly-one **Read-only** unspecified **Value-type** AnyResource **Representation** Either **Range** any **Description** Used to indicate the current state of an Availability Component. It is expected that this will be a resource reference to a definition of a valid target type on the service provider.
`oslc-availability:desiredState`	**Occurs** zero-or-one **Read-only** FALSE **Value-type** AnyResource **Representation** Either **Range** any **Description** Used to indicate the desired state of an Availability Component. It is expected that this will be a resource reference to a definition of a valid target type on the service provider.
`oslc-availability:mttf`	**Occurs** zero-or-one **Read-only** unspecified **Value-type** AnyResource **Representation** Either **Range** any **Description** Mean Time To Failure (MTTF), the average elapsed time between failures of the resource during operation. The Mean Time Between Failure (MTBF) can be calculated by a consumer as the sum of *oslc-availability:mttf* and *oslc-availability:mttr*. It is likely that the target resource will be an `ems:Measure` but that is not necessarily the case.
`oslc-availability:mttr`	**Occurs** zero-or-one **Read-only** unspecified **Value-type** AnyResource **Representation** Either **Range** any **Description** Mean Time To Repair (MTTR), the average

		elapsed time until the resource is repaired after a failure/ the actual measured RTO. The Mean Time Between Failure (MTBF) can be calculated by a consumer as the sum of *oslc-availability:mttf* and *oslc-availability:mttr*. It is likely that the target resource will be an `ems:Measure` but that is not necessarily the case.

Resource: AvailabilityGroup

- **Name:** `AvailabilityGroup`
- **Description:** An *AvailabilityGroup* groups several *AvailabilityResources*, that are in a special relationship to each other in the context of (high) availability.
- **Type URI** `http://open-services.net/ns/availability#AvailabilityGroup`

AvailabilityGroup Properties

Prefixed Name	***Details***	
OSLC Core: Common Properties.		
`rdf:type`	**Occurs**	zero-or-many
	Read-only	unspecified
	Value-type	Resource
	Representation	Reference
	Range	any
	Description	The resource type URIs.
`dcterms:contributor`	**Occurs**	zero-or-many
	Read-only	unspecified
	Value-type	AnyResource
	Representation	Either
	Range	any
	Description	Contributor or contributors to this resource (reference: Dublin Core), that may in some form have influence to this resource, for example change its state. It is likely that the target resource will be an `foaf:Person` but that is not necessarily the case. Another example is an `auto:AutomationPlan`.
`dcterms:creator`	**Occurs**	zero-or-many
	Read-only	unspecified

	Value-type AnyResource **Representation** Either **Range** any **Description** Creator or creators of resource (reference: Dublin Core). It is likely that the target resource will be an `foaf:Person` but that is not necessarily the case. For example the execution of an `oslc_auto:AutomationPlan` can also be responsible for the creation of an availability resource
`dcterms:created`	**Occurs** exactly-one **Read-only** True **Value-type** DateTime **Representation** Inline **Range** any **Description** Timestamp of resource creation (reference: Dublin Core).
`dcterms:description`	**Occurs** zero-or-one **Read-only** unspecified **Value-type** XMLLiteral **Representation** Inline **Range** any **Description** Descriptive text (reference: Dublin Core) about the resource represented as rich text in XHTML content. SHOULD include only content that is valid and suitable inside an XHTML element.
`dcterms:identifier`	**Occurs** exactly-one **Read-only** True **Value-type** String **Representation** Inline **Range** any **Description** A unique identifier for a resource. Assigned by the service provider when a resource is created. Not intended for end-user display.
`oslc:instanceShape`	**Occurs** zero-or-one **Read-only** True **Value-type** Resource **Representation** Reference **Range** `oslc:ResourceShape`

	Description — Resource Shape that provides hints as to resource property value-types and allowed values.
`dcterms:modified`	**Occurs** — zero-or-one **Read-only** — True **Value-type** — DateTime **Representation** — Inline **Range** — any **Description** — Timestamp of latest resource modification (reference: Dublin Core). Updates to the members of this group are not tracked by this property.
`oslc:serviceProvider`	**Occurs** — zero-or-many **Read-only** — True **Value-type** — Resource **Representation** — Reference **Range** — `oslc:ServiceProvider` **Description** — The scope of a resource is a link to the resource's OSLC Service Provider.
`dcterms:subject`	**Occurs** — zero-or-many **Read-only** — unspecified **Value-type** — String **Representation** — Inline **Range** — any **Description** — Tag or keyword for a resource. Each occurrence of a dc:subject property denotes an additional tag for the resource.
`dcterms:title`	**Occurs** — exactly-one **Read-only** — unspecified **Value-type** — XMLLiteral **Representation** — Inline **Range** — any **Description** — A name given to the resource (reference: Dublin Core). If unique, it can be used to identify different automation resources.
`rdf:type`	**Occurs** — zero-or-many **Read-only** — unspecified **Value-type** — Resource

	Representation Reference **Range** any **Description** The resource type URIs.
OSLC AvailabilityGroup: Start of additional properties.	
`oslc-availability:memberListModified`	**Occurs** zero-or-one **Read-only** True **Value-type** DateTime **Representation** Inline **Range** any **Description** Timestamp of latest modification of the member list of this group, (reference: Dublin Core).
`oslc-availability:numberActiveMembers`	**Occurs** zero-or-one **Read-only** TRUE **Value-type** Integer **Representation** Inline **Range** >= 0 **Description** Number of currently active member resources in this group.
`oslc-availability:SLA`	**Occurs** zero-or-many **Read-only** Unspecified **Value-type** Resource **Representation** Reference **Range** any **Description** Reference to the Service Level Agreement (SLA) for this group. This can be a String or even a reference to a document, containing the Service Level Agreement. How such a SLA is modelled in detail, is not part of this specification.

Resource: RedundancyGroup

- **Name:** `RedundancyGroup`
- **Description:** A RedundancyGroup is a group of AvailabilityResources, redundant to each other for providing high availability.
  Note: A resource of type RedundancyGroup is likely also of type AvailabilityGroup.
- **Type URI** `http://open-services.net/ns/availability#RedundancyGroup`

ReplicationGroup Properties

<table>
<tr><th>Prefixed Name</th><th>Details</th></tr>
<tr><td colspan="2">OSLC Core: Common Properties.</td></tr>
<tr><td>rdf:type</td><td>
<table>
<tr><td>Occurs</td><td>zero-or-many</td></tr>
<tr><td>Read-only</td><td>unspecified</td></tr>
<tr><td>Value-type</td><td>Resource</td></tr>
<tr><td>Representation</td><td>Reference</td></tr>
<tr><td>Range</td><td>any</td></tr>
<tr><td>Description</td><td>The resource type URIs.</td></tr>
</table>
</td></tr>
<tr><td colspan="2">OSLC RedundancyGroup: Start of additional properties.</td></tr>
<tr><td>oslc-availability:maxActiveMembers</td><td>
<table>
<tr><td>Occurs</td><td>zero-or-one</td></tr>
<tr><td>Read-only</td><td>unspecified</td></tr>
<tr><td>Value-type</td><td>Integer</td></tr>
<tr><td>Representation</td><td>Inline</td></tr>
<tr><td>Range</td><td>>= 0</td></tr>
<tr><td>Description</td><td>Number of members in this group (likely oslc-availability:AvailabilityResources), that are allowed to be active. Excessing this value implies a problematic state.</td></tr>
</table>
</td></tr>
<tr><td>oslc-availability:minActiveMembers</td><td>
<table>
<tr><td>Occurs</td><td>zero-or-one</td></tr>
<tr><td>Read-only</td><td>unspecified</td></tr>
<tr><td>Value-type</td><td>Integer</td></tr>
<tr><td>Representation</td><td>Inline</td></tr>
<tr><td>Range</td><td>>= 0</td></tr>
<tr><td>Description</td><td>Minimum number ofmembers in this group (likely oslc-availability:AvailabilityResources), that should be active. Underrunning this value implies a problematic state.</td></tr>
</table>
</td></tr>
<tr><td>oslc-availability:redundancyCapability</td><td>
<table>
<tr><td>Occurs</td><td>zero-or-one</td></tr>
<tr><td>Read-only</td><td>TRUE</td></tr>
<tr><td>Value-type</td><td>Integer</td></tr>
<tr><td>Representation</td><td>Inline</td></tr>
<tr><td>Range</td><td>>= 0</td></tr>
<tr><td>Description</td><td>An indication of level of redundancy provided by this group.
Calculated by the formula # group members – minActiveMembers – # offline/ failed members.</td></tr>
</table>
</td></tr>
</table>

`oslc-availability:synchronizationType`	**Occurs** zero-or-one **Read-only** Unspecified **Value-type** AnyResource **Representation** Either **Range** any **Description** Information about how this group's members synchronize each other. It is expected that this will be a resource reference to a definition of a valid target type on the service provider.

Resource: RedundancyMember

- **Name:** `RedundancyMember`
- **Description:** A RedundancyMember is a member of a RedundancyGroup.
  Note: A resource of type RedundancyMember is likely also of type AvailabilityResource.
- **Type URI** `http://open-services.net/ns/availability#RedundancyMember`

ReplicationGroup Properties

Prefixed Name	***Details***
OSLC Core: Common Properties.	
`rdf:type`	**Occurs** zero-or-many **Read-only** unspecified **Value-type** Resource **Representation** Reference **Range** any **Description** The resource type URIs.
OSLC RedundancyMember: Start of additional properties.	
`oslc-availability:redundancyRole`	**Occurs** zero-or-one **Read-only** unspecified **Value-type** AnyResource **Representation** Either **Range** any **Description** Specifies the role of a resource in the context of Availability in terms of redundancy, for example if the resource is a master or a slave component. It is expected that this will be a resource reference to a definition of a valid target type

		on the service provider.

OSLC Actions and Availability

Accordant to the OSLC Actions 2.0 specification, Actions provide “a means of advertising actions (or operations) that can be performed on (or in the context of) a specific resource”.

In the context of the Availability specification, Actions are very likely to be used to change the *oslc-availability:AvailabilityCondition* of an *oslc-availability:AvailabilityResource*. If such a resource represents for example a software system, a typical action may to start or stop it.

Condition action type

This specification defines the RDF class *oslc-availability:ConditionAction*, as an `rdfs:subClassOf` `oslc:Action`, with the meaning that any action of this type **MUST** have the semantics of changing the condition of a resource in the context of Availability. It is likely that the execution of such an action changes the states of a resource's *oslc-availability:AvailabilityCondition*, but this **MAY** not be the case.

Profiles

The profile a provider of this specification must support, depends on the providers capability to support synchronous requests. (Also see the sub-domains section about Availability providers, `http://open-services.net/ns/availability#Sub-Domains`)

All other suggested profiles in this specification are suggested for implementation, but not enforced. Providers can additionally implement other profiles as well.

Profile: Create a HTTP request with an oslc-availability:AvailabilityCondition as request body

As described in http://open-services.net/wiki/core/Actions-2.0/#pattern-resource-shape

A client can change the condition of a resource in the context of Availability by sending a HTTP request with an oslc-availability:AvailabilityCondition as body, that represents the new condition of the resource. The result of the action's execution is a http:StatusCode.

If a provider is capable to execute actions synchronously, it **MUST** support this profile. The action's binding SHOULD specify `oslc:usage` = `oslc:default` in this case.

Profile: Create an Automation Request

As described in http://open-services.net/wiki/core/Actions-2.0/#profile_automation_request and http://open-services.net/wiki/automation/OSLC-Automation-Specification-Version-2.1/

To change the condition of a resource, the client creates an *oslc_auto:AutomationRequest* to execute an *oslc_auto:AutomationPlan*, handling all the necessary steps to change the condition of the resource. It is very likely, that as a side effect the oslc-availability:AvailabilityCondition of the

resource will be changed.

If a provider executes actions asynchronously, it **MUST** support this profile. The action's binding SHOULD specify `oslc:usage` = `oslc:default` in this case. (If a provider supports both, synchronous and asynchronous execution of actions, it **MUST** support both profiles. It can then use the `oslc:usage` property to tell clients, what is the preferred way to execute actions.)

Profile: Use delegated UI dialog for immediate execution

As described in http://open-services.net/wiki/core/Actions-2.0/#pattern-immed-dialog

The client displays a delegated UI dialog to a user to perform an action immediately, that will change the condition of a resource in the context of Availability.

A provider **MAY** support this profile.

Availability Service Provider Capabilities

Lifetime of Availability Resources

An OSLC Availability service provider is generally assumed to represent the availability of a (computer-) system. Especially in a software system, the lifetime of single components can be very short: They can be stopped and/ or deleted etc. Likewise new components can be created at any time. Therefore a client should never make assumptions about the existence of components. The next time it queries for it, it may already by disappeared or its state has completely changed. Clients will always query a snapshot of an availability resource as these are subjects to change.

Availability Provider Sub-Domains

An instance of an OSLC Availability service provider might provide services for one or more particular availability sub-domains (e.g. automated or manual availability). Availability service providers MAY declare sub-domain information in the Service Provider document by specifying a sub-domain value in the `oslc:usage` attribute on the `oslc:Service` resource in the Service Provider document. Valid sub-domain values are:

- **http://open-services.net/ns/availability#Automated**: Indicates that the Availability resources and its condition of the related service provider are controlled by an automation software. Hence the execution of an *oslc-availability:ConditionAction*, to change the condition of an Availability resource, is asynchronously. It is very likely that such an service provider also implements the OSLC Automation Specification and therefore provides oslc_auto:AutomationPlans and oslc_auto:AutomationRequests for handling its Availability resources.
- **http://open-services.net/ns/availability#Manual**: Indicates that the related service provider controls the Availability resources and its condition manually (without an automation software). The execution of an *oslc-availability:ConditionAction*, to change the condition of an Availability resource, is synchronously.

An Availability service provider which is a general-purpose automation provider, or a provider not

wanting to provide a sub-domain should provide an `oslc:usage` value of **http://open-services.net/ns/availability**. If no `oslc:usage` attribute indicating a sub-domain is present, the default is assumed to be **http://open-services.net/ns/availability**.

Resource Shapes

OSLC Availability service providers **MAY** support Resource Shapes as defined in OSLC Core Specification Appendix A

Service Provider Resource

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

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

It is **RECOMMENDED** that OSLC Availability service providers provide a `oslc:serviceProvider` property for their defined resources that will be the URI to a Service Provider Resource.

Creation Factories

If an OSLC Availability service provider supports the creation of resources, there **MUST** be at least one `oslc:creationFactory` entry in the Services definition.

See HTTP Method support table for further clarification on support for HTTP methods and media types for each OSLC Availability resource.

Query Capabilities

OSLC Availability service providers **SHOULD** have at least one `oslc:queryCapability` entry in the its Services definition that allows a client to query *AvailabilityResources*.

The Query Capability **MUST** support these OSLC query parameters and **MAY** support others:

- `oslc:where`
- `oslc:select`

If shape information is NOT present with the Query Capability, service providers **SHOULD** use the default properties defined in OSLC Core RDF/XML Examples to contain the result.

Selective Property Values

OSLC Availability providers **SHOULD** support the `oslc.properties` syntax for selective property value retrieval when a resource is accessible via its resource URI.

Delegated UIs

OSLC Availability service providers support the selection and creation of Availability resources as defined by Delegated UIs in OSLC Core.

The service providers supports requirements for delegated UIs is as follows:

Availability Resource	Selection	Creation
AvailabilityResource	SHOULD	MAY
AvailabilityCondition	SHOULD	---
AvailabilityGroup	SHOULD	MAY
RedundancyGroup	SHOULD	MAY
RedundancyMember	SHOULD	MAY

Properties

OSLC Availability service providers can identify several properties (like states, synchronization type or redundancy role) using references to property values in the OSLC Availability vocabulary or to property values that are not in the Availability vocabulary (i.e. in the service provider's own vocabulary). It is expected that these property values will be URI references, but inline resources defining these property values are also valid.

State Properties

Additional property values for `oslc-availability:desiredState` are:

- `http://open-services.net/ns/availability#Offline` - used to indicate an availability resource is offline/ not available/ not active.
- `http://open-services.net/ns/availability#Online` - used to indicate an availability resource is online/ available/ active.

Additional property values for `oslc-availability:currentState` are:

- `http://open-services.net/ns/availability#Unknown` - used to indicate that the state of an availability resource is unknown.
- `http://open-services.net/ns/availability#Offline` - used to indicate that the availability resource is offline/ not available/ not active.
- `http://open-services.net/ns/availability#Online` - used to indicate that the availability resource is online/ available/ active.
- `http://open-services.net/ns/availability#Starting` - used to indicate that the availability resource is transforming into the online state.
- `http://open-services.net/ns/availability#Stopping` - used to indicate that the availability resource is transforming from the online into the offline state.
- `http://open-services.net/ns/availability#Degraded` - used to indicate that the availability resource is available but has a minor problem, probably making it not not working as expected.
- `http://open-services.net/ns/availability#Problem` - used to indicate that the availability resource is available but has a serious problem, probably making it unusable.

Additional property values for `oslc-availability:compoundState` are:

- `http://open-services.net/ns/availability#Satisfactory`- used to indicate that the resource's desired and observed statuses are corresponding; no further

automation or operator activity is required.

- `http://open-services.net/ns/availability#Problem` - used to indicate that there is a problem with this resource that cannot be solved automatically. The resource is unusable at the moment. Operator intervention is required.
- `http://open-services.net/ns/availability#Inhibited` - used to indicate that the resource is not in its desired state because of a problem with a supporting resource.
- `http://open-services.net/ns/availability#Degraded` for an Availability Group used to indicate that it doesn't match the expected grade of availability. For Availability Resources, it can mean that the resource is Starting or Stopping, or that the resource is suffering from a performance or throughput problem.

Redundancy properties

Additional property values for `oslc-availability:redundancyRole` are:

- `http://open-services.net/ns/availability#Primary` - used to indicate that a RedundancyMember is the primary (master) resource of a Redundancy Group.
- `http://open-services.net/ns/availability#Secondary` - used to indicate that a RedundancyMember is a secondary (slave) resource of a Redundancy Group.

Additional property values for `oslc-availability:synchronizationType` are:

- `http://open-services.net/ns/availability#Asynchronous` - used to indicate that the members of an RedundancyGroup synchronize each other asynchronously.
- `http://open-services.net/ns/availability#Synchronous` - used to indicate that the members of an RedundancyGroup synchronize each other synchronously.

Availability Service Provider HTTP method support

Support for all HTTP methods in the compliance table is not required for all Availability resources. The following table summarizes the requirements for each resource type, HTTP method and for each media type.

Resource	RDF/XML	XML	JSON	OSLC (Compact)	HTML	Unspecified
AvailabilityResource						
GET	MUST	MAY	SHOULD	SHOULD	SHOULD	N/A
PUT	MAY	MAY	MAY	N/A	N/A	N/A
POST	MAY	MAY	MAY	N/A	N/A	N/A
DELETE	N/A	N/A	N/A	N/A	N/A	MAY
AvailabilityCondition						
GET	MUST	MAY	SHOULD	SHOULD	SHOULD	N/A
PUT	N/A	N/A	N/A	N/A	N/A	N/A

Resource	RDF/XML	XML	JSON	OSLC (Compact)	HTML	Unspecified
POST	N/A	N/A	N/A	N/A	N/A	N/A
DELETE	N/A	N/A	N/A	N/A	N/A	N/A
AvailabilityGroup						
GET	MUST	MAY	SHOULD	SHOULD	SHOULD	N/A
PUT	MAY	MAY	MAY	N/A	N/A	N/A
POST	MAY	MAY	MAY	N/A	N/A	N/A
DELETE	N/A	N/A	N/A	N/A	N/A	MAY
RedundancyMember						
GET	MUST	MAY	SHOULD	SHOULD	SHOULD	N/A
PUT	MAY	MAY	MAY	N/A	N/A	N/A
POST	MAY	MAY	MAY	N/A	N/A	N/A
DELETE	N/A	N/A	N/A	N/A	N/A	MAY
RedundancyMember						
GET	MUST	MAY	SHOULD	SHOULD	SHOULD	N/A
PUT	MAY	MAY	MAY	N/A	N/A	N/A
POST	MAY	MAY	MAY	N/A	N/A	N/A
DELETE	N/A	N/A	N/A	N/A	N/A	MAY

OSLC Availability service providers **SHOULD** support deletion of any resources for which it allows creation.

Availability Specification Guidance

This section is informative.

Updating the condition of an AvailabilityResource

The condition of an Availability Resource is represented by its Availability Condition, referenced through the `oslc_asset:state` attribute. Some providers may allow their consumers, to update the condition of an Availability Resource. For example if it represents a running software system, they may want to stop it or vice versa.

To update an Availability Resource's condition, the client in general will need to execute an `oslc-availability:ConditionAction`. The necessary action is likely to be referenced by the resource's `oslc:action` attribute.

But first the client need to check the current condition of the resource, to see if an update is really necessary. This is done by sending a HTTP GET request to fetch the Availability Resource's `oslc-availability:AvailabilityCondition` object and analyse the `oslc-`

`availability:currentState` and `oslc-availability:compoundState` attributes.

If the client still needs to update the Availability Resource's condition, it must execute the relevant `oslc-availability:ConditionAction`. Depending on the providers capability to support synchronous or asynchronous requests (also see Availability Provider Sub-Domains), a different action profile needs to be used:

- **Synchronous requests**: The profile “Create a HTTP request with an oslc-availability:AvailabilityCondition as request body” is necessary: The client will post an `oslc-availability:AvailabilityCondition` object, with the `oslc-availability:desiredState` attribute set to the required state. The provider will then try to update the condition accordingly and immediately respond with a HTTP status code to signal the success or failure.
- **Asynchronous requests**: The profile “Create an Automation Request” is necessary: The client will create an `oslc_auto:AutomationRequest`, with an `oslc_auto:InputParameter` providing the new value for the `oslc-availability:desiredState`. The provider will then try to update the condition accordingly but asynchronously. The client needs to poll for the `oslc_auto:AutomationResult` to see when its request has completed and if it was successful.

Appendix A: Samples

(this section is informative)

Appendix B: Resource Shapes

(this section is informative)

Appendix C: Future Prospects

Replication

A possible enhancement for future versions of this specification is the consideration of (data) replication. Replication may also be seen as an Availability topic and is not covered yet by any other public OSLC specification.

The introduction of a concept to reflect replication would also imply the introduction of a Recovery Point Objective (RPO) -goal for Availability Components and its actual measurement for the Availability Conditions.

Appendix D: Notices and References

Contributors

We thank Martin Pain (IBM) and John Arwe (IBM) a lot for their help and advice during the elaboration of this specification.