OSLC Availability Specification Draft 0.~~6~~8

Status: 0.~~6~~8 Draft Specification - ~~1~~20 ~~July~~August 2014

This Version

- OSLC Availability Specification Version 0.~~5~~8

Latest Version

- OSLC Availability Specification Version 0.~~5~~8

Previous Version

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

Authors

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

Table of Contents

License

Notation and Conventions

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

Introduction

(this section is informative)

This specification builds on the OSLC Core Specification to define the resources and operations supported by an Open Services for Lifecycle Collaboration (OSLC) Availability provider. This specification is closely coupled with the OSLC Automation Specification, providing the possibility to modify the state of resources.

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.~~but it can be used for other types of systems.

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~~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	SHOULD	Core	OSLC service providers SHOULD support HTTP

Requirement	*Level*	*Origin(s)*	*Meaning*
Representations			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
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

Requirement	*Level*	*Origin(s)*	*Meaning*
			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.

Namespace prefix used	Namespace URI	Usage
`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**~~**SHOULD** support an [X]HTML representation and a user interface (UI) preview as defined by UI Preview Guidance

For HTTP PUT/POST request formats for 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:action rdf:ID=”action1”
            rdf:resource=”http://example.com/actions/1” />
      <oslc_rm:affectedBy rdf:ID="link1"
            rdf:resource="http://example.com/plans/123" />
```

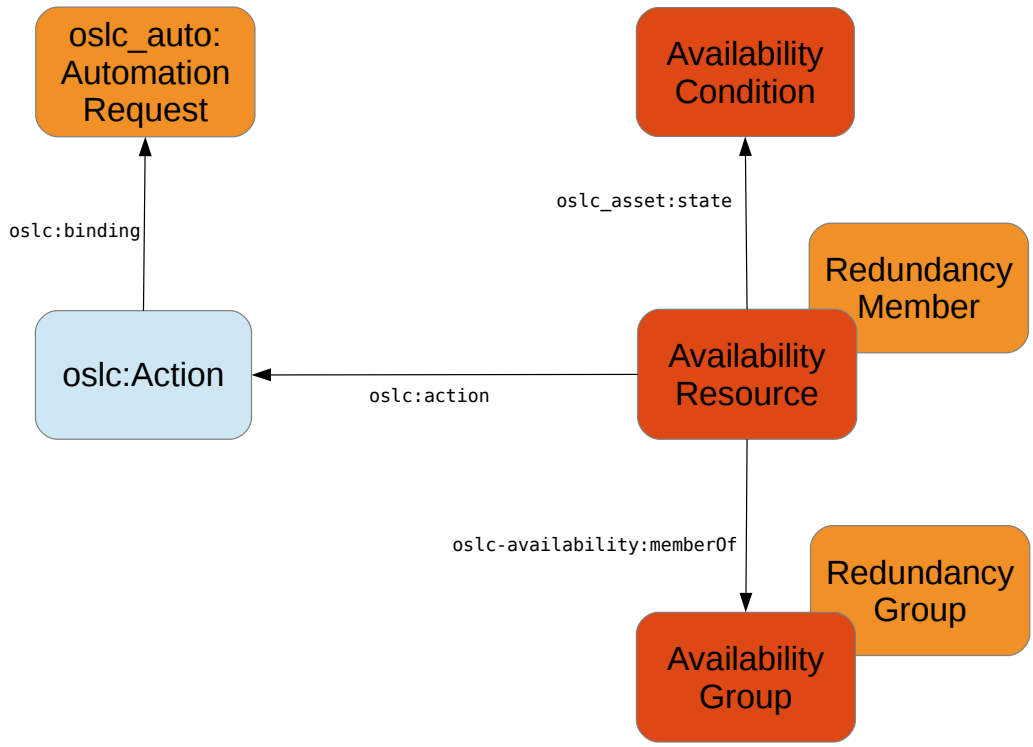
```
    </oslc-availability-auto:AvailabilityResource>

    <rdf:Description rdf:about="#actionlink1">
      <dcterms:title>Condition Action Automation Plan 1231: Pet Shop App
production buildStarting the Pet Shop App production webserver</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. It is RECOMMENDED that Aany additional properties SHOULD exist in their own unique namespace and do 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**) SHOULDMUST 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.

Note: If the value-type of a property from this specification's resources is DateTime, it SHOULD contain the time zone information.

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`

AvailabilityResource~~Component~~ **Properties**

Prefixed Name	*Details*	
OSLC Core: Common Properties		
`oslc:action`	**Occurs**	zero-or-many
	Read-only	unspecified
	Value-type	Resource
	Representation	Either
	Range	any
	Description	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:`~~`Change`~~`ConditionAction`~~ ~~–~~ but that is not necessarily the case~~.~~; ~~when it is an `oslc-availability:ConditionAction`, it~~

		~~will have at least one binding since it is currently available.~~
dcterms:created	**Occurs**	zero-or-one
	Read-only	True
	Value-type	DateTime
	Representation	Inline
	Range	any
	Description	Timestamp of resource creation (reference: Dublin Core)
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: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**	zero-or-many~~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-one~~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	~~Any~~Resource
Representation	Either
Range	`oslc-availability:AvailabilityCondition`
Description	Detailed information about this ~~the~~Availabiliry Resource's ~~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~~Either **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). ~~resources.automationIf unique, it can be used to identify different~~
rdf:type	**Occurs** one~~zero~~-or-many **Read-only** unspecified **Value-type** Resource **Representation** Reference **Range** any **Description** The resource type URIs.
OSLC AvailabilityResource: Start of additional properties including relationship properties.	
oslc-availability:memberOf	**Occurs** zero-or-many **Read-only** unspecified **Value-type** Resource **Representation** Reference **Range** any **Description** 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 oslc-availability:**Availability–Group** (~~or and/~~ or its sub-type oslc-availability:**Redundancy–Group**), but this is not necessarily the case.

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(s) to the characteristic of this Availability Condition.~~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:description`	**Occurs**	zero-or-one
	Read-only	unspecified
	Value-type	XMLLiteral
	Representation	~~Inline~~Either
	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**	zero-or-many~~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:modified	
Occurs	exactly-one
Read-only	True
Value-type	DateTime
Representation	Inline
Range	any
Description	Timestamp of latest resource modification (reference: Dublin Core): The point in time this condition has occurred.

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	one~~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:~~compound~~consistentState	
Occurs	one~~zero~~-or-many~~one~~
Read-only	unspecified
Value-type	AnyResource
Representation	Either

	Range	any
	Description	Used to indicate if the Availability Resource is in a consistent state, based on values defined by the service provider. It is very likely that this property is used as a compound state of the `oslc-availability:currentState` and `oslc-availability:desiredState`, but this is no necessarily the case. ~~`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~~zero-or-many
	Read-only	unspecified
	Value-type	AnyResource
	Representation	Either
	Range	any
	Description	Used to indicate the current state of an Availability ~~Component~~Resource based on values defined by the service provider. ~~-~~ If not set, it is assumed that the Availability Resource's current state is unknown. 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~~many
	Read-only	False
	Value-type	AnyResource
	Representation	Either
	Range	any
	Description	Used to indicate the desired state of an Availability ~~Component~~Resource based on values defined by the service provider. ~~-~~ If not set, it is assumed that the Availabilty Resource's `oslc-availability:currentState` is the desired state, It is expected that this will be a resource reference to a definition of a valid target type on the service provider.

Resource: AvailabilityGroup

- **Name:** `AvailabilityGroup`
- **Description:** An *AvailabilityGroup* groups several *AvailabilityResources*, that are in somea special relationship to each other in the context of (high) availability. The exact nature of this special relationship is not constrained by the definition of AvailabilityGroup, but it might be conveyed and/or constrained by other types that the resource has.
  It is possible, that an *AvailabilityGroup* is an *AvailabilityResource* itself (`rdfs:subClassOf: AvailabilityResource`), but that is not necessarily the case.
- **Type URI** `http://open-services.net/ns/availability#AvailabilityGroup`

AvailabilityGroup 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 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** zero-or-many~~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 states/ condition of this group's members ~~of this group~~ are not tracked

	by this property, but changes in group membership are tracked by this property.~~.~~

oslc:action	

Occurs	**zero-or-many**
Read-only	unspecified
Value-type	Resource
Representation	Either
Range	any
Description	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 AvailabilityGroup is to add or remove a member, likely an AvailabilityResource, to or from it. 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:MembershipAction`, but that is not necessarily the case. Note: If an AvailabilityGroup is also of type `oslc-availability:AvailabilityResource`, it is very likely that target resources of this property can also be `oslc-availability:ConditionActions`, but that is not necessarily the case.

oslc:serviceProvider	

Occurs	zero-or-one~~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** — one~~zero~~-or-many **Read-only** — unspecified **Value-type** — Resource **Representation** — Reference **Range** — any **Description** — The resource type URIs.

Resource: RedundancyGroup

- **rdfs:subClassOf: `AvailabilityGroup`**
- **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

Prefixed Name	*Details*
OSLC RedundancyGroup: Start of additional properties.	
`oslc-availability:max~~Act ive~~Available~~~~Members`	**Occurs** — zero-or-one **Read-only** — unspecified **Value-type** — Integer **Representation** — Inline **Range** — any~~>= 0~~ **Description** — Maximum ~~N~~number of members in this group (likely **`oslc-`**

		availability:AvailabilityResources), ~~that~~allowed to be ~~activeare allowed to be~~ available. ~~Excessing this value implies a problematic state.~~The numeric value SHOULD be greater than or equal to 0. Together with oslc-availability:minAvailableMembers, this property allows to indicate the capacity of redundancy.
oslc-availability:min~~Act ive~~Available~~Members~~	**Occurs**	zero-or-one
	Read-only	unspecified
	Value-type	Integer
	Representation	Inline
	Range	any~~>= 0~~
	Description	Minimum number of members in this group (likely oslc-availability:AvailabilityResources), ~~that should~~allowed to be ~~active~~available. ~~Underrunning this value implies a problematic state.~~ The numeric value SHOULD be greater than or equal to 0. Together with oslc-availability:maxAvailableMembers, this property allows to indicate the capacity of redundancy.

Resource: RedundancyMember

- **rdfs:subClassOf: AvailabilityResource**
- **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 RedundancyMember: Start of additional properties.		
oslc-availability:redundancyRole	**Occurs**	~~zero-or~~-one-or-many
	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, based on values defined by the service provider. It is expected that this will be a resource reference to a definition of a valid target type on the service provider.

Resource: ConditionAction

- **rdfs:subClassOf: `oslc:Action`**
- **Name:** `ConditionAction`
- **Description:** An Action that changes the condition of an resource on its execution.
- **Type URI** `http://open-services.net/ns/availability#ConditionAction`

`ConditionAction` Properties

Prefixed Name	*Details*	
OSLC ConditionAction: Start of additional properties.		
`oslc-availability:desiredState`	**Occurs**	one-or-many
	Read-only	True
	Value-type	AnyResource
	Representation	Either
	Range	any
	Description	Used to indicate the desired state, an Availability Resource's condition will be changed to by executing this action. It is expected that this will be a resource reference to a definition of a valid target type on the service provider.

Resource: MembershipAction

- **rdfs:subClassOf: `oslc:Action`**
- **Name:** `MembershipAction`
- **Description:** An Action that changes the membership of an resource to a group on its execution.
- **Type URI** `http://open-`

services.net/ns/availability#MembershipAction

ConditionMembershipAction Properties

Prefixed Name	*Details*	
OSLC MembershipAction: Start of additional properties.		
oslc-availability:membershipOperation	**Occurs**	exactly-one
	Read-only	True
	Value-type	AnyResource
	Representation	Either
	Range	any
	Description	Used to indicate the membership operation, the execution of this action will perform. It is very likely that this operation will be adding a new member to a group or removing an existing member from a group. 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

Accord~~ant~~ing 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, the preferred way to change the the oslc-availability:AvailabilityCondition of an oslc-availability:AvailabilityResource is the usage of 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.

Actions are also the preferred way to manage the membership of an oslc-availability:AvailabilityResource to an oslc-availability:AvailabilityGroup.

Condition action type

This specification defines the RDF class **oslc-availability:ConditionAction**, as an **rdfs:subClassOf** of **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. The execution of such an action SHOULD change the states of an oslc-availability:AvailabilityCondition (oslc-availability:consistentState and/ or oslc-availability:currentState).~~*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.*~~

The state, the condition of a resource will be changed into, is specified by the Condition Action's `oslc-availability:desiredState` property.

Membership action type

This specification defines the RDF class `oslc-availability:MembershipAction`, as an `rdfs:subClassOf` of `oslc:Action`, with the meaning that any action of this type MUST have the semantics of managing the membership of a group in the context of Availability. The execution of such an action SHOULD add or remove a member (an `oslc-availability:AvailabilityResource`) to or from an `oslc-availability:AvailabilityGroup`. Since the membership between resources and a group is represented with the `oslc-availability:memberOf` property, the execution of a Membership action SHOULD alter this property at the Availability Resource.

The membership operation, that will be performed by an Membership Action (e.g. add or remove a member), is specified by the Membership Action's `oslc-availability:membershipOperation` property.

The URI of the `oslc-availability:AvailabilityResource`, that is added or removed from an `oslc-availability:AvailabilityGroup`, must be provided as additional parameter during execution of this action.

~~Profiles~~OSLC Actions Specification Profiles

The OSLC Action Specification **profile** a provider of this specification ~~must~~MUST support ~~,~~ depends on the value in the `oslc:usage` attribute on the `oslc:Service` resource, see section **Availability Provider Sub-Domains** of this specification.

If the Availability Resources are automated (**`http://open-services.net/ns/availability#Automated`**, see section *Availability Provider Sub-Domains* in this specification), the Action Profile *Create an Automation* Request MUST be supported.
If not, the Action Profile *POST RDF described by a OSLC Resource Shape to the Action resource* MUST be supported instead.

~~) sub-domains section about Availability providers, `http://open-services.net/ns/availability#Sub-Domains` see the. (Also providers capability to support synchronous requests~~

The profiles suggested in this section are already defined in the *OSLC Actions Specification*, see OSLC Actions Specification Profiles.

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

Profile: POST RDF described by a OSLC Resource Shape to the Action resource ~~*Create a HTTP request with an oslc-availability:AvailabilityCondition as request body*~~

As described in the *OSLC Action Specification*, section *Specification profile definitions*.

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

Service providers with an `oslc:usage` of `http://open-services.net/ns/core#default` MUST support this profile. The action's binding SHOULD specify `oslc:usage = oslc:default` in this case.

Changing a group membership:

A client can change the membership of a group in the context of Availability by sending a HTTP request with an `oslc-availability:AvailabilityResource` as body. The resource in the body represents the Availability Resource with a modified `oslc-availability:memberOf`-relationship. The result of the action's execution is a http:StatusCode.

Changing a resource's condition:

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

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

Profile: Create an Automation Request

As described in the *OSLC Action Specification*, section *Specification profile definitions* and in the *OSLC Automation Specification*. ~~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/~~

Service providers with an `oslc:usage` of **`http://open-services.net/ns/availability#Automated`** MUST support this profile. The action's binding SHOULD specify `oslc:usage = oslc:default` in this case.

Changing a group membership:

A client can change the membership of a resource to a group in the context of Availability by creating an `oslc_auto:AutomationRequest` to execute an `oslc_auto:AutomationPlan`, that is responsible for handle all necessary steps in an automated environment.

An `oslc-availability:AvailabilityGroup` list all executable membership operations as `oslc-availability:MembershipActions` through its `oslc:Action` property. The Membership Action's `oslc-availability:membershipOperation` property tells the client what operation will be performed by this action.

The Action bindings `http:body` of the desired membership operation contains the `oslc_auto:AutomationRequest`, the client needs to create to perform this operation. The group, for which this operation is performed, is identified by the Availability Group, from which the client has fetched this Membership Action. The Availability Resource, that will be added or removed,

must be identified by the client within the `oslc_auto:AutomationRequest` as `oslc_auto:inputParameter`, an `oslc_auto:ParameterInstance` with `oslc:name` of value `oslc-availability:AvailabilityResource`.

The client can verify that the operation was successful by consulting the `oslc_auto:AutomationResult` object, that is the result of its Automation Request. (See the OSLC Automation Specification for further details.)

Changing a resource's condition:

A client can change the condition of a resource in the context of Availability by creating an `oslc_auto:AutomationRequest` to execute an `oslc_auto:AutomationPlan`, that is responsible for handle all necessary steps in an automated environment.

An `oslc-availability:AvailabilityResource` lists all possible condition changes, that can be performed on this resource as `oslc-availability:ConditionActions` through its `oslc:Action` property. The Condition Action's `oslc-availability:desiredState` property tells the client what condition change will be performed by this action.

The Action bindings `http:body` of the desired action contains the `oslc_auto:AutomationRequest`, the client needs to create to perform this condition change. The desired state and the Availability Resource's URI are already set as `oslc_auto:inputParameters` in this Automation Request.

The client can verify that the operation was successful by consulting the `oslc_auto:AutomationResult` object, that is the result of its Automation Request. (See the OSLC Automation Specification for further details.)

~~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

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

~~An OSLC Availability service provider is generally assumed to represent the availability of a~~

Availability Provider Sub-Domains

An instance of an OSLC Availability service provider might provide services for one or more particular Aa~~a~~vailability sub-domains (e.g. automated ~~or manual availability~~resources). 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. (See `http://open-services.net/ns/core#Service` for more details.)

Valid sub-domain values are:

- **http://open-services.net/ns/availability#Automated**: Indicates that the Availability R~~r~~esources, including their Availability Conditions, ~~and its condition of the related service provider~~ are controlled by ~~an~~ automation software: As a result, clients SHOULD not be able to manually manage Availability Group membership or interfere with the condition of these resources. Instead they SHOULD use the provided automation mechanisms.
  For a client to be able to do so, such a service provider SHOULD also implement the OSLC Automation Specification and provide `oslc_auto:AutomationPlans` and `oslc_auto:AutomationRequests` for handling its Availability Resources.
  Service providers, implementing this Sub-Domain, MUST support the *Create an Automation Request* Action Profile.
  ~~. 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.ly of an *oslc-availability:ConditionAction*, to change the condition of an Availability resource, is asynchronousexecution Hence the .~~

~~.ly of an *oslc-availability:ConditionAction*, to change the condition of an Availability resource, is synchronousexecutionautomation software). The an **http://open-services.net/ns/availability#Manual**: Indicates that the related service provider controls the Availability resources and its condition manually (without~~

An Availability service provider which is a general-purpose ~~automation~~availability provider, or a provider not wanting to provide a sub-domain should provide an **`oslc:usage`** value of `http://open-services.net/ns/core#default`~~**`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/core#default`**~~**`http://open-services.net/ns/availability`**~~.
Such service providers MUST support the *POST RDF described by a OSLC Resource Shape to the Action resource* Action Profile, to allow a manual update of resources.

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	MAY[1] ~~---~~
AvailabilityGroup	SHOULD	MAY
RedundancyGroup	SHOULD	MAY
RedundancyMember	SHOULD	MAY

Properties

OSLC Availability service providers can identify several properties (like the current state~~s~~ of an Availability Resource or the redundancy role of a member in an Redundancy Group)~~, 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 to property values, but inline resources defining these property values are also valid. Availability service providers MUST use at least one of the property values defined in the OSLC Availability vocabulary in addition to any property values not in the Availability vocabulary.

State Properties

Property values for `oslc-availability:consistentState` are:

- `http://open-services.net/ns/availability#Consistent` - used to indicate an availability resource's state is consistent.
- `http://open-services.net/ns/availability#NotConsistent` - used to indicate an availability resource's state is not consistent.

Property values for `oslc-availability:currentState` are:

- `http://open-services.net/ns/availability#Unknown` - used to indicate that the availability resource's current state is unknown.
- `http://open-services.net/ns/availability#Unavailable` - used to indicate that the availability resource is not available.
- `http://open-services.net/ns/availability#Available` - used to indicate

1 If applicable.

that the availability resource is available.

- `http://open-services.net/ns/availability#InMaintenance` - used to indicate that the availability resource is not available due to maintenance.
- `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 working as expected. Operator intervention may be required.
- `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. Operator intervention is required.

Property values for `oslc-availability:desiredState` are:

- `http://open-services.net/ns/availability#Unavailable` - used to indicate that the availability resource SHOULD be not available, means its `oslc-availability:currentState` SHOULD transform into `http://open-services.net/ns/availability#Unavailable`.
- `http://open-services.net/ns/availability#Available` - used to indicate that the availability resource SHOULD be available, means its `oslc-availability:currentState` SHOULD transform into `http://open-services.net/ns/availability#Available`.
- `http://open-services.net/ns/availability#InMaintenance` - used to indicate that the availability resource SHOULD be not available for maintenance purpose, means its `oslc-availability:currentState` SHOULD transform into `http://open-services.net/ns/availability#InMaintenance`.

~~`available/ active. online/ - used to indicate an availability resource is Online` not available/ not active.~~

~~`http://open-services.net/ns/availability#offline/ - used to indicate an availability resource is Offline`roperty values for `oslc-availability:desiredState` are:~~

~~`http://open-services.net/ns/availability#Additional p`~~

~~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

P~~Additional p~~roperty 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.
- `http://open-services.net/ns/availability#p2p` - used to indicate that a Redundancy Member is a resource of a peer-to-peer network (grouped by 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.~~

Membership properties

Property values for `oslc-availability:membershipOperation` are:

- `http://open-services.net/ns/availability#Add` - used to indicate that a new member is added to a group.
- `http://open-services.net/ns/availability#Remove` - used to indicate that an existing member is removed from a group.

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~~MAY	~~N/A~~MAY	~~N/A~~MAY	N/A	N/A	N/A
POST	~~N/A~~MAY	~~N/A~~MAY	~~N/A~~MAY	N/A	N/A	N/A
DELETE	N/A	N/A	N/A	N/A	N/A	~~N/A~~MAY
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.

Querying and identifying Availability Resources and Availability Groups

Clients can retrieve `oslc-availabiity:AvailabilityResources` and `oslc-availability:AvailabilityGroups` (including their sub-types) by using an Availability Service Provider's Query Capabilities, as described by the OSLC Core Specification.

An Availability Resource's `dcterms:identifier` properties must be unique within the scope of the service provider, as should be their `dcterms:title` properties. Therefore these properties can be used by clients to identify and query for specific Availability Resources. (Also see OSLC Core Specification's Query Syntax.)

If a client is only interested in Availability Groups or even only in Redundancy Groups or Redundancy Members, it can query for them using the `rdf:type` field in the query parameters.

Querying and understanding the condition of an Availability Resource

The condition of a specific `oslc-availability:AvailabilityResource` is represented by its `oslc-availability:AvailabilityCondition`. If a client is interested in the condition of an Availability Resource, it can query this resource's Availability Condition by reading the Availability Resource's `oslc_asset:state` property.

With the Availability Condition, a client can understand the state of the originated Availability Resource, since when this state is in effect and who or what has contributed etc..

Since a service provider must use at least one of the property values, defined in this specification, for the state properties like `oslc-availability:currentState` and `oslc-availability:consistentState`, the client has a fundamental understanding of the resource's condition. If the service provider additionally uses own property values to describe the Availability Condition, it should provide an explanation about how these property values have to be interpreted.

If the Availability Resources of a service provider for example represent server applications (database management system, web server etc.), the client can assume that an `oslc-availability:currentState` with the value `http://open-services.net/ns/availability#Available` means that an application is up and running, while a value of `http://open-services.net/ns/availability#Unavailable` means that it is offline.

Creating and deleting Availability Resources

Creation and deletion of Availability Resources – if provided by the service provider – is done as described in the *OSLC Core Specification*, section *Resource Creation* and *Resource Removal*.

Querying members of an Availability Group and review the overall status

To query all members of an Availability Group, a client must first identify this Availability Group, as described in the section *Querying and identifying Availability Resources and Availability Groups*.

As next step, the client can query all Availability Resources, where the `oslc-availability:memberOf` property contains the identified Availability Group (see OSLC Core Specification's Query Syntax).

To review the group's overall status, the client can read all members `oslc-availability:AvailabilityCondition`, as described in section *Querying and understanding the condition of an Availability Resource*.
If all `oslc-availability:consistenState` properties of the Availability Conditions contain the value `http://open-services.net/ns/availability#Consistent`, the group's overall status is fine. If at least one member's condition of the group contains the value `http://open-services.net/ns/availability#NotConsistent`, the group may need operator intervention and the Availability Condition of this member should be investigated further.

Managing Availability Groups

Some Availability service providers might allow clients to manage an Availability Group's members, so that they can add or remove Availability Resources to or from them. Group management is done using `oslc-availability:MembershipActions`.

The client needs to identify the Availability Group, as described in the section *Querying and identifying Availability Resources and Availability Groups*.

The Availability Group's `oslc:action` property lists all available `oslc-availability:MembershipActions` available for this group.
The Action Profile that must be implemented by an Availability Service Provider, and therefore the recommended interaction pattern for clients, depends on the support of sub-domains:

- Availability Service Providers that implement the `http://open-services.net/ns/availability#Automated` sub-domain, represent automated environments which are using `oslc_auto:AutomationPlans` for their management. They must support the *Create an AutomationRequest*-profile.
  To add or remove a member to a group, the client must select the group's Action with the matching `oslc-availability:membershipOperation` property. This Action's `oslc:binding` refers to the `oslc_auto:AutomationRequest`, the client must create to perform the desired operation. To identify the Availability Resource, that will be added or removed to or from the Availability Group, the client must identify the resource as `oslc_auto:inputParamter`, an `oslc_auto:ParameterInstance` with `oslc:name` of value `oslc-availability:AvailabilityResource`.
- Availability Service Providers that are general purpose Availability providers, must support the *POST RDF described by a OSLC Resource Shape to the Action resource*-profile.
  To update the membership to a group, clients locally update the `oscl-availability:memberOf`-property of an Availability Resource as desired and send this updated resource with a HTTP PUT operation to the Availability Resource's URI, see the *OSLC Core Specification*, section *Resource Update*.

Note that only the relation between the group and the resource is deleted by this operation, neither the Availability Resource, nor the Availability Group is deleted. To delete a resource, see section *Creating and deleting Availability Resources*.

~~Updating~~Change the condition of an AvailabilityResource

Some Availability service providers might allow clients to change an Availability Resource's condition, for example start an unavailable software application. Changing the condition is done using `oslc-availability:ConditionActions`.

The client needs to identify the Availability Resource, as described in the section *Querying and identifying Availability Resources and Availability Groups.*

The Availability Resource's `oslc:action` property lists all available `oslc-availability:ConditionActions` available for this resource.
The Action Profile that must be implemented by an Availability Service Provider, and therefore the recommended interaction pattern for clients, depends on the support of sub-domains:

- Availability Service Providers that implement the `http://open-services.net/ns/availability#Automated` sub-domain, represent automated environments which are using `oslc_auto:AutomationPlans` for their management. They must support the *Create an AutomationRequest*-profile.
  To change the condition of a resource, the client must select the resource's Action with the matching `oslc-availability:desiredState` property (for example `oslc-availability:Available`). This Action's `oslc:binding` refers to the `oslc_auto:AutomationRequest`, the client must create to perform the desired operation.
- Availability Service Providers that are general purpose Availability providers, must support the *POST RDF described by a OSLC Resource Shape to the Action resource*-profile.
  To change the condition of a resource, clients locally update the `oscl-availability:AvailabilityCondition's oscl-availability:desiredState`-property as desired and send this updated resource with a HTTP PUT operation to the Availability Condition's URI, see the *OSLC Core Specification*, section *Resource Update*.
  The Availability Condition can be fetched from the Availability Resource's `oslc_assed:state` property.

~~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.~~Redundancy information

`oslc-availability:RedundancyGroups` are a specific type of `oslc-availability:AvailabilityGroups`, that group redundant resources.

The redundant members can be queried as described in the section *Querying members of an Availability Group and review the overall status*. It is very likely that all members of this group are

oslc-availability:RedundancyMembers. The oslc-availability:redundancyRole specifies the role of such an member within the group, for example if it is the primary member, that should be active per default or the secondary one, that takes over if the primary one fails.

The oslc-availability:minAvailableMembers and oslc-availability:maxAvailableMembers properties of a Redundancy Group provides information about the redundancy goal, that should be achieved by this group, and therefore about the "level" of redundancy. The oslc-availability:minAvailableMembers property indicates the minimum number of resources in the group that must be available, otherwise the goal is not achieved and the Redundancy Group probably cannot provide its service as desired. The oslc-availability:maxAvailableMembers property indicates the maximum number of resources in the group that should be available, for example because the other members should be kept as fail-over members.

If the actual number of group members, that are available, heads toward the oslc-availability:minAvailableMembers value, the level of redundancy is low: If another member fails, the availability:minAvailableMembers value may be underrun and the Redundancy Group may miss its goals.

If the actual number of group members, that are available, heads toward the oslc-availability:maxAvailableMembers value, the level of redundancy is high. Other members may still fail without underrunning the availability:minAvailableMembers value and the Redundancy Group still achieves its goals.

~~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.~~

Typical Scenarios in the context of Availability

This section is informative.

This section gives some example about handling with workloads in the context of Availability.

Workloads are single entities or groups of entities executed on a server for the purpose of fulfilling a particular business value. Examples are started tasks on a z/OS system, a middleware subsystem consisting of several processes / address spaces or even multi-tiered business applications that can span multiple servers.

Listing workloads

Workloads, represented as Availability Resources by an Availability service provider, can be listed as described in the section *Querying and identifying Availability Resources and Availability Groups*.

The workloads status information can be retrieved as described in the section *Querying and understanding the condition of an Availability Resource*.

Starting and stopping workloads

A workload, represented as Availability Resource by an Availability service provider, can be started or stopped as described in the section *Change the condition of an AvailabilityResource*.

A stopped workload can be identified by its `oslc-availability:currentState` value of `http://open-services.net/ns/availability#Unavailable`. (It may have additional other values for this property, specified by the service provider, to provide more information about this condition.) To start the workload, the client must update its `oslc-availability:desiredState` to a value of `http://open-services.net/ns/availability#Available` (additional values, specified by the service provider, may be possible as well, for example to differ between a cold or hot start).

Obtain redundancy information for a workload

A service, provided by a workload may be secured against outages: It is said to be highly available. One possibility to secure a service against outages is redundancy. The workload is covered by redundant components (software systems, hardware etc.), performing the same task. If one of these components fails, the remaining components are still able to provide the service and the operator can repair or replace the failed component.

A workload, that is covered by redundant components is represented as Redundancy Group by an Availability Service provider. The Redundancy Group allows to identify the level of redundancy, it provides. The members of this group, the Redundancy Members, provide the information about their state themselves (section *Querying and understanding the condition of an Availability Resource*). They also provide the information about their role in the context of redundancy: For example a group with two redundant members, one configured as primary member, that is available per default. The other configured as secondary member, available to take over of the primary one fails. See section *Redundancy information* for further details.

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.