OSLC Architecture Management Version 3.0. Part 1: Specification

Project Specification 01
30 September 2021

This stage:

https://docs.oasis-open-projects.org/oslc-op/am/v3.0/ps01/architecture-management-spec.html (Authoritative)
https://docs.oasis-open-projects.org/oslc-op/am/v3.0/ps01/architecture-management-spec.pdf

Previous stage:

https://docs.oasis-open-projects.org/oslc-op/am/v3.0/psd01/architecture-management-spec.html (Authoritative)
https://docs.oasis-open-projects.org/oslc-op/am/v3.0/psd01/architecture-management-spec.pdf
(published as Project Specification Draft on 17 September 2020)

Latest stage:

https://docs.oasis-open-projects.org/oslc-op/am/v3.0/architecture-management-spec.html (Authoritative)
https://docs.oasis-open-projects.org/oslc-op/am/v3.0/architecture-management-spec.pdf

Latest version:

https://open-services.net/spec/am/latest

Latest editor's draft:

https://open-services.net/spec/am/latest-draft

Open Project:

OASIS Open Services for Lifecycle Collaboration (OSLC) OP

Project Chairs:

Jim Amsden (jamsden@us.ibm.com), IBM
Andrii Berezovskyi (andriib@kth.se), KTH

Editor:

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

Additional components:

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

- *OSLC Architecture Management Version 3.0. Part 1: Specification (this document)*. architecture-management-spec.html
- *OSLC Architecture Management Version 3.0. Part 2: Vocabulary*. architecture-management-vocab.html

- *OSLC Architecture Management Version 3.0. Part 3: Constraints*. architecture-management-shapes.html
- *OSLC Architecture Management Version 3.0. Part 4: Machine Readable Vocabulary Terms*. architecture-management-vocab.ttl
- *OSLC Architecture Management Version 3.0. Part 5: Machine Readable Constraints*. architecture-management-shapes.ttl

Related work:

This specification is related to:

- *OSLC Architecture Management Specification Version 2.0*. http://open-services.net/wiki/architecture-management/OSLC-Architecture-Management-Specification-Version-2.0/

RDF Namespaces:

http://open-services.net/ns/core/am#

Abstract:

This specification defines the OSLC Architecture Management domain, a RESTful web services interface for the management of architectural resources and relationships between those and related resources such as product change requests, activities, tasks, requirements or test cases. To support these scenarios, this specification defines a set of HTTP-based RESTful interfaces in terms of HTTP methods: GET, POST, PUT and DELETE, as well as HTTP response codes, content type handling and resource formats.

Status:

This document was last revised or approved by the OASIS Open Services for Lifecycle Collaboration (OSLC) OP on the above date. The level of approval is also listed above. Check the "Latest stage" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Open Project are listed at https://open-services.net/about/.

Comments on this work can be provided by opening issues in the project repository or by sending email to the project's public comment list oslc-op@lists.oasis-open-projects.org.

Note that any machine-readable content (Computer Language Definitions) declared Normative for this Work Product is provided in separate plain text files. In the event of a discrepancy between any such plain text file and display content in the Work Product's prose narrative document(s), the content in the separate plain text file prevails.

Citation format:

When referencing this specification the following citation format should be used:
[OSLC-AM-3.0-Part1]
OSLC Architecture Management Version 3.0. Part 1: Specification. Edited by Jim Amsden. 30 September 2021. OASIS Project Specification 01. https://docs.oasis-open-projects.org/oslc-op/am/v3.0/ps01/architecture-management-spec.html.
Latest stage: https://docs.oasis-open-projects.org/oslc-op/am/v3.0/architecture-management-spec.html.

Notices

Table of Contents

1. Introduction

This section is non-normative.

This specification defines a RESTful web services interface for the Architecture Management (AM) domain. This domain addresses the management of product design artifacts including models, and relationships with other resources such as requirements, testing resources and change requests. To support these scenarios, this specification defines a set of HTTP-based RESTful interfaces in terms of HTTP methods: GET, POST, PUT and DELETE, HTTP response codes, content type handling and resource formats..

The intent of this specification is to define the capabilities needed to support integration scenarios defined by the Architecture Management working group and not to provide a comprehensive interface to Architecture Management. The resource formats and operations may not match exactly the native artifacts supported by architecture management AM Servers but are intended to be compatible with them. The approach to supporting these scenarios is to delegate operations, as driven by service provider contributed user interfaces, as much as possible and not require a service provider to expose its complete data model and application logic.

This specification is a [OSLCCore3] compliant specification, and as such most of its content are references to [OSLCCore3].

1.1 Terminology

Resource

An artifact used in the Application Lifecycle Management (ALM) space. A resource is directly addressable with an absolute URL.

Architecture Management Resource (AMR)

Directly addressable resources of some domain/notation (i.e. UML, BMPN, ER) that represent an abstraction of some behavior or construct of a system under development. An AMR maintains its identity after refactoring. In the semantic web, an AMR might correspond to a graph that is an instance of some vocabulary or micro-theory.

Link

A logical relationship from one resource to another resource. An OSLC AM Link is uni-directional. The subject (source) of a link represents the resource that "knows about" and is referencing another resource (target). The type of relationship is given by a predicate URI (link type). In semantic web terminology, a link would correspond to an RDF statement with a subject (source), a predicate (type) and object (target). The predicate could be defined by property in an RDF schema.

Link type (LT)

A URI that represents the type of a link. In semantic web terminology it is the predicate of an RDF triple. It clarifies the type of relationship between two resources. Link Type URIs may be defined locally, within the OSLC, or externally (i.e. Dublin Core terms). Link types could be defined in RDF Schemas.

Link type Resource (LTR)

A resource that contains human consumable information about a Link Type, like its human readable name and description. The resource is managed by the AM provider. The information may be about a Link Type in a different domain (i.e. Dublin Core Terms or OWL). The main use of an LTR is for clients who want to build a UI for users that clearly labels potential link types.

AM Client

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

AM Server

A server implementing the OSLC Architecture Management domain specifications. OSLC AM clients consume services provided by AM Servers. The use of the terms Client and Server are intended to distinguish typical consumers and providers of OSLC resources in a distributed environment based on REST. A particular application component could be a client for some OSLC domain services and a server for the same or another domain.

1.2 References

1.2.1 Normative references

[OSLCCore2]

Dave Johnson; S. Speicher. *OSLC Core Specification 2.0*. https://open-services.net/. Finalized. URL: https://archive.open-services.net/bin/view/Main/OslcCoreSpecification

[OSLCCore3]

Jim Amsden; S. Speicher. *OSLC Core Version 3.0. Part 1: Overview*. OASIS. Project Specification Draft. URL: https://docs.oasis-open-projects.org/oslc-op/core/v3.0/oslc-core.html

[OSLCPreview]

OSLC Core Version 3.0. Part 3: Resource Preview0. OASIS. Project Specification Draft. URL: https://docs.oasis-open-projects.org/oslc-op/core/v3.0/resource-preview.html

[OSLCShapes]

Arthur Ryman; Jim Amsden. *OSLC Core Version 3.0. Part 6: Resource Shape*. OASIS. Project Specification Draft. URL: https://docs.oasis-open-projects.org/oslc-op/core/v3.0/resource-shape.html

[RFC2119]

S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF, March 1997. Best Current Practice. URL: https://www.rfc-editor.org/rfc/rfc2119

[RFC8174]

B. Leiba. *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*. IETF, May 2017. Best Current Practice. URL: https://www.rfc-editor.org/rfc/rfc8174

1.2.2 Informative references

[LDPPatch]

Linked Data Patch Format. http://www.w3.org/. Working Group Note. URL: http://www.w3.org/TR/ldpatch/

1.3 Typographical Conventions and Use of RFC Terms

As well as sections marked as non-normative, all authoring guidelines, diagrams, examples, and notes in this specification are non-normative. Everything else in this specification is normative.

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this specification are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Base Requirements

The following sub-sections define the mandatory and optional requirements for an OSLC Architecture Management (OSLC AM) server.

2.1 Base Compliance

This specification is based on [OSLCCore3]. OSLC AM servers **MUST** be compliant with both the core specification, **MUST** follow all the mandatory requirements in the normative sections of this specification, and **SHOULD** follow all the guidelines and recommendations in both these specifications. [am-1]

An OSLC AM server **MUST** implement the domain vocabulary defined in OSLC Architecture Management Version 2.1. Part 2: Vocabulary [am-2]

The following table summarizes the requirements from OSLC Core Specification as well as some additional requirements specific to the AM domain. Note that this specification further restricts some of the requirements from the OSLC Core Specification. See the previous sections in this specification or the OSLC Core Specification to get further details on each of these requirements.

Requirement	Meaning
Absolute URIs	AM Servers **MUST** use absolute URIs for all references to resources by properties [am-3]
Unknown properties and content	AM Servers **MAY** ignore unknown content and AM clients **MUST** preserve unknown content. AM Servers **MAY** discard such properties and continue the POST or PUT operation without warning to the client. [am-4]
Resource Operations	AM Servers **MUST** support resource operations via standard HTTP operations [am-5]
Update and Delete	AM Servers **SHOULD** support resource modifications with standard HTTP PUT and DELETE methods. AM Servers **MAY** limit modifications [am-6]
HTTP If-Match use	AM Servers supporting update and delete of resources **MUST** support the standard HTTP If-Match header in PUT and DELETE for concurrency protection of resources. [am-7]
Resource Paging	AM Servers **MAY** provide paging for resources but only when specifically requested by clients [am-8]
Partial Resource Representations	AM Servers **MAY** support requests for a subset of a resource's properties via the oslc.properties URL parameter retrieval via HTTP GET [am-9]
Partial Update	AM Servers **MAY** support partial update of resources via the oslc.properties URL parameter retrieval via HTTP PUT and or using [LDPPatch]. [am-10]
Discovery	AM Servers **MAY** provide a Service Provider Catalog, **MUST** provide a Service Provider resource, and **MAY** provide other forms of discovery described in [OSLCCore3]. [am-11]
Creation Factories	AM Servers **MAY** provide creation factories for resource formats that it supports. AM Servers **MAY** support creation factories for OSLC AM defined resources formatted as application/rdf+xml. AM Servers **MAY** support creation factories for other formats, and indicate such creation factories with a non-default identifier in the oslc:usage property of the creation factory definition in the service provider document [am-12]

Requirement	Meaning
Query Capabilities	AM Servers **MUST** provide query capabilities on oslc_am:Resource resources to enable clients to query for resources. AM Servers **SHOULD** support a query interface for oslc_am:LinkType resources that support a GET for all LinkType resources. Such a GET does not require any simple query syntax parameters. AM Servers **MAY** support the full query syntax for LinkType resources. [am-13]
Query Syntax	OSLC query capabilities **MUST** support the OSLC Core Query Syntax [am-14]
Delegated Dialogs	AM Services **SHOULD** offer selection delegated dialogs and **MAY** offer creation delegated dialogs specified via service provider resource [am-15]
Resource Preview	AM Services **SHOULD** offer resource previews for resources that may be referenced by other resources [am-16]
Authentication	AM Services **SHOULD** follow the recommendations for Authentication specified in [OSLCCore3] [am-17]
Error Responses	AM Servers **SHOULD** provide error responses using OSLC Core defined error formats [am-18]
RDF/XML Representations	AM Servers **MUST** support RDF/XML representations for OSLC Defined Resources [am-19]
XML Representations	AM Servers **MUST** support XML representations that conform to the OSLC Core Guidelines for XML [am-20]
JSON Representations	AM Servers **MAY** support JSON representations; those which do **MUST** conform to the OSLC Core Guidelines for JSON [am-21]
HTML Representations	AM Servers **MAY** provide HTML representations for GET requests [am-22]

2.2 Specification Versioning

This specification follows the specification version guidelines given in [OSLCCore3].

2.3 Namespaces

In addition to the namespace URIs and namespace prefixes `oslc`, `rdf`, `dcterms` and `foaf` defined in [OSLCCore3], OSLC AM defines the namespace URI of `http://open-services.net/ns/am#` with a preferred namespace prefix of `oslc_am`.

2.4 Resource Formats

In addition to the requirements for resource representations in [OSLCCore3], this section outlines further refinements and restrictions.

For HTTP GET/PUT/POST requests on all OSLC AM and OSLC Core defined resource types,

- AM Servers **MUST** support RDF/XML representations with media-type `application/rdf+xml`. AM Clients **SHOULD** be prepared to deal with any valid RDF/XML document. [am-23]
- AM Servers **MUST** support XML representations with media-type `application/xml`. The XML representations **MUST** follow the guidelines outlined in the OSLC Core Representations Guidance to maintain compatibility with [OSLCCore2]. [am-24]
- AM Servers **MAY** support JSON representations with media-type `application/json`. The JSON representations **MUST** follow the guidelines outlined in the OSLC Core Representations Guidance to maintain compatibility with [OSLCCore2]. [am-25]

2.5 Resource Operations

For compatibility with OSLC 2.0, OSLC AM Servers **MAY** accept the OSLC Core Version header (OSLC-Core-Version: 2.0) in any HTTP request as specified in [OSLCCore3], and return an OSLC AM 2.0 representation (including the OSLC-Core-Version: 2.0 header). If the OSLC Core Version header is absent on a request, or has some undefined value, the OSLC AM Server **MUST** return an AM 3.0 representation. [am-26]

Since OSLC 3.0 is a compatible superset of OSLC 2.0, an AM 3.0 representation may also be an AM 2.0 representation, even if the OSLC Core Version header is absent.

OSLC AM Servers **MUST** support HTTP GET requests on Architecture Management Resources (AMR), with an Accept header of `application/rdf+xml`, and return the RDF/XML representation of the resource. [am-27]

OSLC AM Servers **SHOULD** support HTTP GET requests on Architecture Management Resources (AMR), with an Accept header of an HTML type (`application/html`, `application/xhtml`), and return either an HTML/XHTML representation of the resource or redirect the client to another URL that can (i.e. 302 Redirect). [am-28]

OSLC AM Servers **SHOULD** support HTTP GET requests for user interface (UI) preview of Architecture Management Resources (AMR) as defined by [OSLCPreview]. [am-29]

OSLC AM Servers **SHOULD** support resource modifications on Architecture Management Resources (AMR) with standard HTTP PUT and DELETE methods. AM Servers **MAY** limit modifications in any way they want. For example a service provider may limit updates to resources to simple link properties of link types already defined in the provider. Modification methods **MUST** use the If-Match header for concurrency management. Providers **MAY** discard such properties and continue a PUT operation without warning to the client. [am-30]

OSLC AM Servers **SHOULD** support resource modifications on LinkType Resources (LTR) with standard HTTP PUT and DELETE methods. AM Servers **MAY** limit modifications in any way they want. For example a service provider may not support additional properties. Modification methods **SHOULD** use the If-Match header for concurrency management. [am-31]

2.6 Authentication

See [OSLCCore3], OSLC AM puts no additional constraints on authentication.

2.7 Error Responses

See [OSLCCore3], OSLC AM puts no additional constraints on error responses

2.8 Pagination

OSLC AM Servers **SHOULD** support pagination of query results and **MAY** support pagination of a single resource's properties as defined by [OSLCCore3]. [am-32]

2.9 Requesting and Updating Properties

2.9.1 Requesting a Subset of Properties

An OSLC AM server **MAY** support the `oslc.properties` URL query parameter on an HTTP GET request on individual resource request or a collection of resources by query. If the `oslc.properties` query parameter is omitted on the request, then all resource properties **MUST** be provided in the response. [am-33]

2.9.2 Updating a Subset of Properties

An OSLC AM client **MAY** request that a subset of a resource's properties be updated by identifying those properties to be modified using the `oslc.properties` URL parameter on a HTTP PUT request. [am-34]

2.9.3 Updating Multi-Valued Properties

An OSLC AM Server **MAY** support updating a subset of a resource's properties by using the [LDPPatch] `PATCH` method. [am-35]

For compatibility with [OSLCCore2], an AM Server **MAY** also support partial update by identifying those properties to be modified using the `oslc.properties` URL parameter on a HTTP PUT request. [am-36]

If the parameter `oslc.properties` contains a valid resource property on the request that is not provided in the content, the server **MUST** set the resource's property to a null or empty value. If the parameter `oslc.properties` contains an invalid resource property, then a `409 Conflict` **MUST** be returned. [am-37]

3. Vocabulary Terms and Constraints

OSLC Architecture Management Resources 2.1 Defines the vocabulary terms and constraints for OSLC Change Management resources. These terms and constraints are specified according to [OSLCCore3].

4. AM Server Capabilities

4.1 Resource Shapes

OSLC AM servers **SHOULD** support Resource Shapes as defined in [OSLCShapes]. [am-38]

4.2 Service Provider Resources

OSLC AM Servers **MUST** provide a ServiceProvider Resource that can be retrieved at a implementation dependent URI. [am-39]

OSLC AM Servers **MUST** provide a ServiceProviderCatalog Resource that can be retrieved at a implementation dependent URI. [am-40]

OSLC AM Servers **MUST** provide an `oslc:serviceProvider` property for their defined resources that will be the URI to a ServiceProvider Resource. This does not prevent AM Servers from providing multiple servie provider properties with different values, if the service provider supports multiple OSLC domain specifications, and the resource is applicable to multiple domains. [am-41]

OSLC AM Servers **MUST** supply a value of `http://open-services.net/ns/am#` for the property `oslc:domain` on either `oslc:ServiceProvider` or `oslc:ServiceProviderCatalog` resources. [am-42]

4.3 Creation Factories

OSLC AM Servers **MAY** support CreationFactories as defined by [OSLCCore3]. [am-43]

OSLC AM Servers **MAY** discard properties it does not recognize and continue the POST operation without warning to the client. The returned resource will contain the accepted properties (and server generated properties like the dcterms:identifer) so clients will be able to confirm if required what was accepted. [am-44]

If OSLC AM Servers support the creation of resources from the OSLC defined `oslc_am:Resource` format, there **MUST** be at least one Creation Factory entry in the Services definition, and its `oslc:usage` property **MUST** be set to `http://open-services/ns/core#default`. The `oslc:resourceType` **MUST** be set to `http://open-services.net/ns/am#Resource`. [am-45]

If OSLC AM Servers support the creation of resources from a resource other than `oslc_am:Resource`, there **MUST** be a separate creation services definition whose `oslc:usage` property **MUST NOT** be set to `http://open-services/ns/core#default`. [am-46]

4.4 Query Capabilities

OSLC AM Servers **SHOULD** support the Query Capabilities as defined by [OSLCCore3] for both `oslc_am:Resource` and `oslc_am:LinkType` resources. [am-47]

If the service provider supports query capability for `oslc_am:Resource` resources, it **MUST** support the following query parameters: [am-48]

- `oslc.where`
- `oslc.searchTerms`

OSLC AM Servers **SHOULD** support query capability for `oslc_am:LinkType` resources. If supported then AM Servers **MUST** support a simple GET without any query parameters that returns all link type resources. AM Servers **SHOULD** support the full OSLC query syntax. [am-49]

4.5 Delegated UIs

OSLC AM Servers **SHOULD** support the selection of resources by delegated selection dialogs as defined by [OSLCCore3]. [am-50]

OSLC AM Servers **MAY** support the creation of resources by delegated creation dialogs as defined by [OSLCCore3]. [am-51]

In `oslc:Dialog` elements, the two optional child elements; `oslc:hintWidth` and `oslc:hintHeight` specify the suggested size of the dialog or frame to render the HTML content in. Expected size values are defined by CSS length units. [am-52]

5. Conformance

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

Clause Number	Requirement
am-1	This specification is based on [OSLCCore3]. OSLC AM servers **MUST** be compliant with both the core specification, **MUST** follow all the mandatory requirements in the normative sections of this specification, and **SHOULD** follow all the guidelines and recommendations in both these specifications.
am-2	An OSLC AM server **MUST** implement the domain vocabulary defined in OSLC Architecture Management Version 2.1. Part 2: Vocabulary
am-3	AM Servers **MUST** use absolute URIs for all references to resources by properties
am-4	AM Servers **MAY** ignore unknown content and AM clients **MUST** preserve unknown content. AM Servers **MAY** discard such properties and continue the POST or PUT operation without warning to the client.
am-5	AM Servers **MUST** support resource operations via standard HTTP operations
am-6	AM Servers **SHOULD** support resource modifications with standard HTTP PUT and DELETE methods. AM Servers **MAY** limit modifications
am-7	AM Servers supporting update and delete of resources **MUST** support the standard HTTP If-Match header in PUT and DELETE for concurrency protection of resources.
am-8	AM Servers **MAY** provide paging for resources but only when specifically requested by clients
am-9	AM Servers **MAY** support requests for a subset of a resource's properties via the oslc.properties URL parameter retrieval via HTTP GET
am-10	AM Servers **MAY** support partial update of resources via the oslc.properties URL parameter retrieval via HTTP PUT and or using [LDPPatch].
am-11	AM Servers **MAY** provide a Service Provider Catalog, **MUST** provide a Service Provider resource, and **MAY** provide other forms of discovery described in [OSLCCore3].
am-12	AM Servers **MAY** provide creation factories for resource formats that it supports. AM Servers **MAY** support creation factories for OSLC AM defined resources formatted as application/rdf+xml. AM Servers **MAY** support creation factories for other formats, and indicate such creation factories with a non-default identifier in the oslc:usage property of the creation factory definition in the service provider document
am-13	AM Servers **MUST** provide query capabilities on oslc_am:Resource resources to enable clients to query for resources. AM Servers **SHOULD** support a query interface for oslc_am:LinkType resources that support a GET for all LinkType resources. Such a GET does not require any simple query syntax parameters. AM Servers **MAY** support the full query syntax for LinkType resources.
am-14	OSLC query capabilities **MUST** support the OSLC Core Query Syntax
am-15	AM Services **SHOULD** offer selection delegated dialogs and **MAY** offer creation delegated dialogs specified via service provider resource
am-16	AM Services **SHOULD** offer resource previews for resources that may be referenced by other resources
am-17	AM Services **SHOULD** follow the recommendations for Authentication specified in [OSLCCore3]
am-18	AM Servers **SHOULD** provide error responses using OSLC Core defined error formats
am-19	AM Servers **MUST** support RDF/XML representations for OSLC Defined Resources
am-20	AM Servers **MUST** support XML representations that conform to the OSLC Core Guidelines for XML
am-21	AM Servers **MAY** support JSON representations; those which do **MUST** conform to the OSLC Core Guidelines for JSON
am-22	AM Servers **MAY** provide HTML representations for GET requests
am-23	AM Servers **MUST** support RDF/XML representations with media-type `application/rdf+xml`. AM Clients **SHOULD** be prepared to deal with any valid RDF/XML document.

Clause Number	Requirement
am-24	AM Servers **MUST** support XML representations with media-type `application/xml`. The XML representations **MUST** follow the guidelines outlined in the OSLC Core Representations Guidance to maintain compatibility with [OSLCCore2].
am-25	AM Servers **MAY** support JSON representations with media-type `application/json`. The JSON representations **MUST** follow the guidelines outlined in the OSLC Core Representations Guidance to maintain compatibility with [OSLCCore2].
am-26	For compatibility with OSLC 2.0, OSLC AM Servers **MAY** accept the OSLC Core Version header (OSLC-Core-Version: 2.0) in any HTTP request as specified in [OSLCCore3], and return an OSLC AM 2.0 representation (including the OSLC-Core-Version: 2.0 header). If the OSLC Core Version header is absent on a request, or has some undefined value, the OSLC AM Server **MUST** return an AM 3.0 representation.
am-27	OSLC AM Servers **MUST** support HTTP GET requests on Architecture Management Resources (AMR), with an Accept header of `application/rdf+xml`, and return the RDF/XML representation of the resource.
am-28	OSLC AM Servers **SHOULD** support HTTP GET requests on Architecture Management Resources (AMR), with an Accept header of an HTML type (`application/html`, `application/xhtml`), and return either an HTML/XHTML representation of the resource or redirect the client to another URL that can (i.e. 302 Redirect).
am-29	OSLC AM Servers **SHOULD** support HTTP GET requests for user interface (UI) preview of Architecture Management Resources (AMR) as defined by [OSLCPreview].
am-30	OSLC AM Servers **SHOULD** support resource modifications on Architecture Management Resources (AMR) with standard HTTP PUT and DELETE methods. AM Servers **MAY** limit modifications in any way they want. For example a service provider may limit updates to resources to simple link properties of link types already defined in the provider. Modification methods **MUST** use the If-Match header for concurrency management. Providers **MAY** discard such properties and continue a PUT operation without warning to the client.
am-31	OSLC AM Servers **SHOULD** support resource modifications on LinkType Resources (LTR) with standard HTTP PUT and DELETE methods. AM Servers **MAY** limit modifications in any way they want. For example a service provider may not support additional properties. Modification methods **SHOULD** use the If-Match header for concurrency management.
am-32	OSLC AM Servers **SHOULD** support pagination of query results and **MAY** support pagination of a single resource's properties as defined by [OSLCCore3].
am-33	An OSLC AM server **MAY** support the `oslc.properties` URL query parameter on an HTTP GET request on individual resource request or a collection of resources by query. If the `oslc.properties` query parameter is omitted on the request, then all resource properties **MUST** be provided in the response.
am-34	An OSLC AM client **MAY** request that a subset of a resource's properties be updated by identifying those properties to be modified using the `oslc.properties` URL parameter on a HTTP PUT request.
am-35	An OSLC AM Server **MAY** support updating a subset of a resource's properties by using the [LDPPatch] `PATCH` method.
am-36	For compatibility with [OSLCCore2], an AM Server **MAY** also support partial update by identifying those properties to be modified using the `oslc.properties` URL parameter on a HTTP PUT request.
am-37	If the parameter `oslc.properties` contains a valid resource property on the request that is not provided in the content, the server **MUST** set the resource's property to a null or empty value. If the parameter `oslc.properties` contains an invalid resource property, then a `409 Conflict` **MUST** be returned.
am-38	OSLC AM servers **SHOULD** support Resource Shapes as defined in [OSLCShapes].
am-39	OSLC AM Servers **MUST** provide a ServiceProvider Resource that can be retrieved at a implementation dependent URI.
am-40	OSLC AM Servers **MUST** provide a ServiceProviderCatalog Resource that can be retrieved at a implementation dependent URI.

Clause Number	Requirement
am-41	OSLC AM Servers **MUST** provide an `oslc:serviceProvider` property for their defined resources that will be the URI to a ServiceProvider Resource. This does not prevent AM Servers from providing multiple servie provider properties with different values, if the service provider supports multiple OSLC domain specifications, and the resource is applicable to multiple domains.
am-42	OSLC AM Servers **MUST** supply a value of `http://open-services.net/ns/am#` for the property `oslc:domain` on either `oslc:ServiceProvider` or `oslc:ServiceProviderCatalog` resources.
am-43	OSLC AM Servers **MAY** support CreationFactories as defined by [OSLCCore3].
am-44	OSLC AM Servers **MAY** discard properties it does not recognize and continue the POST operation without warning to the client. The returned resource will contain the accepted properties (and server generated properties like the dcterms:identifer) so clients will be able to confirm if required what was accepted.
am-45	If OSLC AM Servers support the creation of resources from the OSLC defined `oslc_am:Resource` format, there **MUST** be at least one Creation Factory entry in the Services definition, and its `oslc:usage` property **MUST** be set to `http://open-services/ns/core#default`. The `oslc:resourceType` **MUST** be set to `http://open-services.net/ns/am#Resource`.
am-46	If OSLC AM Servers support the creation of resources from a resource other than `oslc_am:Resource`, there **MUST** be a separate creation services definition whose `oslc:usage` property **MUST NOT** be set to `http://open-services/ns/core#default`.
am-47	OSLC AM Servers **SHOULD** support the Query Capabilities as defined by [OSLCCore3] for both `oslc_am:Resource` and `oslc_am:LinkType` resources.
am-48	If the service provider supports query capability for `oslc_am:Resource` resources, it **MUST** support the following query parameters:
am-49	OSLC AM Servers **SHOULD** support query capability for `oslc_am:LinkType` resources. If supported then AM Servers **MUST** support a simple GET without any query parameters that returns all link type resources. AM Servers **SHOULD** support the full OSLC query syntax.
am-50	OSLC AM Servers **SHOULD** support the selection of resources by delegated selection dialogs as defined by [OSLCCore3].
am-51	OSLC AM Servers **MAY** support the creation of resources by delegated creation dialogs as defined by [OSLCCore3].
am-52	In `oslc:Dialog` elements, the two optional child elements; `oslc:hintWidth` and `oslc:hintHeight` specify the suggested size of the dialog or frame to render the HTML content in. Expected size values are defined by CSS length units.

Appendix A. Samples

See OSLC Architecture Management 2.0 Appendix A: Samples

Appendix B. Acknowledgements

This section is non-normative.

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

James Amsden, IBM (Editor)
Chris Armstrong, Armstrong Process Group
Andy Berner, IBM
Scott Bosworth, IBM
Jim Conallen, IBM
Derry Davis, Accenture
Brenda Ellis, Northrop Grumman Corporation
Ian Green, IBM
Jonathan Harclerode, Accenture
Simon Helsen, IBM
Clyde Icuspit, IBM
Wally Mclaughlin, Armstrong Process Group
Thomas Picolli, IBM
Vishy Ramaswamy, IBM
Ren Renganathan. Citi Bank
Nick Crossley, IBM