



---

# OSLC Core v3.0 Changes Version 1.0

## Project Note 01

16 December 2021

### This stage:

<https://docs.oasis-open-projects.org/oslc-op/core-3.0-changes/v1.0/pn01/core-3.0-changes.html> (Authoritative)  
<https://docs.oasis-open-projects.org/oslc-op/core-3.0-changes/v1.0/pn01/core-3.0-changes.pdf>

### Previous stage:

N/A

### Latest stage:

<https://docs.oasis-open-projects.org/oslc-op/core-3.0-changes/v1.0/core-3.0-changes.html> (Authoritative)  
<https://docs.oasis-open-projects.org/oslc-op/core-3.0-changes/v1.0/core-3.0-changes.pdf>

### Latest editor's draft:

<https://oslc-op.github.io/oslc-specs/notes/core-3.0-changes/core-3.0-changes.html>

### Open Project:

[OASIS Open Services for Lifecycle Collaboration \(OSLC\) OP](#)

### Project Chairs:

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

Andrii Berezovskyi ([andriib@kth.se](mailto:andriib@kth.se)), [KTH](#)

### Editor:

David Honey ([DavidHoney@uk.ibm.com](mailto:DavidHoney@uk.ibm.com)), [IBM](#)

### Related work:

This specification is related to:

- *OSLC Core Version 3.0. Part 1: Overview*. Edited by Jim Amsden and Andrii Berezovskyi. 26 August 2021. OASIS Standard: <https://docs.oasis-open-projects.org/oslc-op/core/v3.0/os/oslc-core.html>

### Abstract:

This note will serve as a guide for existing OSLC 2.0 users to learn what's new and navigate around the changes in the specifications.

### Status:

This is a Non-Standards Track Work Product. The patent provisions of the OASIS IPR Policy do not apply.

## Non-Standards Track Work Product

This document was last revised or approved by the Project Governing Board of 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](mailto:oslc-op@lists.oasis-open-projects.org).

### Citation format:

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

#### **[OSLC-Core-Changes-v1.0]**

*OSLC Core v3.0 Changes Version 1.0*. Edited by David Honey. 16 December 2021. OASIS Project Note 01.

<https://docs.oasis-open-projects.org/oslc-op/core-3.0-changes/v1.0/pn01/core-3.0-changes.html>. Latest stage:

<https://docs.oasis-open-projects.org/oslc-op/core-3.0-changes/v1.0/core-3.0-changes.html>.

## Notices

Copyright © OASIS Open 2021. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full [Policy](#) may be found at the OASIS website.

This specification is published under the [Attribution 4.0 International \(CC BY 4.0\)](#). Portions of this specification are also provided under the [Apache License 2.0](#).

All contributions made to this project have been made under the [OASIS Contributor License Agreement \(CLA\)](#).

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Open Project or OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

The name "OASIS" is a trademark of [OASIS](#), the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <https://www.oasis-open.org/policies-guidelines/trademark/> for above guidance.

## Table of Contents

1. [Introduction](#)
  2. [Resource representations](#)
  3. [OSLC Discovery](#)
  4. [Creation factories](#)
  5. [Delegated dialogs](#)
  6. [Query capabilities](#)
  7. [Prefixes](#)
  8. [Resource Paging](#)
  9. [Resource preview](#)
  10. [Shapes and vocabularies](#)
  11. [Authentication](#)
  12. [CORS protocol](#)
- Appendix A. [Acknowledgements](#)

## 1. Introduction

We are happy to announce a publication of OSLC 3.0, the next generation of the core OSLC specification, as the OASIS Standard. The specification in full can be viewed freely [online](#) and this note will serve as a guide for existing OSLC 2.0 users to learn what's new and navigate around the changes in the specifications.

The OSLC Core 2.0 evolution from 1.0 strengthened its definition of the use of RESTful HTTP web services and adopted an existing standard-based approach to representing a data model that was flexible and had multiple representation formats, namely RDF. A primary goal of OSLC Core 3.0 was upwards compatibility with OSLC 2.0, preserving the investment in existing implementations. Most OSLC Core 3.0 changes are optional and OSLC 2.0 servers will not be mandated to adopt them to be compliant with 3.0. In most cases, existing OSLC 2.0 clients should be able to consume OSLC 3.0 servers with no changes. This should result in a less disruptive change from 2.0 to 3.0. This document provides a summary of the changes in OSLC Core 3.0 with respect to OSLC Core 2.0.

From the standardization point of view, there have been two major changes in OSLC 3.0 that further the openness of OSLC and strengthen its integration value. First, the specification development was moved to OASIS, a vendor-neutral standardization body with a proven track record of assisting in the publication of standards that are widely adopted and further standardized by international bodies like ISO (e.g. [MQTT](#), [ODF \(ISO/IEC 26300-1:2015\)](#)). Throughout the development of the specifications, our member section at OASIS featured as many as [16 members](#) before executing a transition to an Open Project that has drastically lowered the bar to [entry](#). Second, the OSLC 3.0 specification is now aligned with the [Linked Data Platform 1.0](#), a W3C Recommendation, that was adopted by projects such as [Solid](#), focused on decentralizing data storage and spearheaded by Sir Tim Berners-Lee, the founding father of the Web.

## 2. Resource representations

OSLC Core 2.0 mandates support for RDF/XML (`application/rdf+xml`) and made support for other RDF formats optional. OSLC Core 3.0 requires that servers support at least one RDF format, and recommends servers support RDF/XML, Turtle (`text/turtle`), and JSON-LD (`application/ld+json`), and use standard HTTP content negotiation in choosing which RDF format to return. Turtle is the preferred format used for examples in OSLC 3.0 specifications. Clients that specify both RDF/XML and Turtle formats in an `Accept` request header are likely to operate with both OSLC 2.0 and 3.0 servers without change. New clients that *only* accept Turtle might not work with older OSLC 2.0 servers that might only support RDF/XML.

### 3. OSLC Discovery

The discovery mechanisms defined in OSLC Core 2.0 for discovering OSLC service provider catalogs, service providers, services, creation factories, selection dialogs, and query capabilities continue to be supported in OSLC Core 3.0. If new OSLC 3.0 server implementations continue to support these discovery mechanisms, they should be consumable by existing OSLC 2.0 clients.

OSLC Core 3.0 extends OSLC 2.0 Discovery to allow servers to optionally support the use of `OPTIONS` and `Link` headers as an alternative mechanism to discover creation factories, selection dialogs, and resource preview. OSLC Core 3.0 further specifies well-known URIs registered with accordance to the [RFC 8615](#) for probing the existence of OSLC Service Provider Catalogs as well as Jazz RootServices documents at well-known URIs.

## 4. Creation factories

Creation factories in OSLC Core 3.0 are upwards compatible with Core 2.0 and existing OSLC 2.0 clients should be able to consume creation factories in an OSLC 3.0 server.



## 5. Delegated dialogs

OSLC Core 3.0 continues to support the use of delegated dialogs for creation and/or selection, and existing OSLC 2.0 clients should be able to consume these without change where these are declared in OSLC services.

## 6. Query capabilities

In OSLC Core 2.0, the use of query capabilities was defined by a section on discovery and a separate document on query syntax. In OSLC Core 3.0, there is a separate OSLC Query 3.0 specification that covers usage and query syntax, and describes the changes from, and compatibility with, query in OSLC Core 2.0. OSLC 2.0 clients should be able to consume query capabilities in an OSLC 3.0 server without change.

## 7. Prefixes

OSLC Core 2.0 did not define a minimum set of prefixes that OSLC servers should predefine and declare in services. OSLC Core 3.0 defines a set of recommended predefined prefixes. Existing servers are not required to implement this to be compliant with OSLC Core 3.0, but it is recommended they do so for convenience to clients.

## 8. Resource Paging

Resource paging is unchanged in OSLC Core 3.0 with respect to Core 2.0.

## 9. Resource preview

In OSLC Core 2.0, clients can request a compact representation of a resource using `Accept=application/x-oslcompact+xml`. This continues to be supported in OSLC Core 3.0. OSLC 3.0 servers that support this functionality will work with OSLC 2.0 clients.

OSLC Core 3.0 also supports a new way of discovering and consuming compact data for resource preview. An OSLC 3.0 server can include a `Link` header where the link relation is `http://open-services.net/ns/core#Compact` and the target URI is the URI of a compact resource. This also supports getting a compact representation in JSON. OSLC 2.0 clients won't be aware of the new `Link` header and hence will not be impacted by this change.

## 10. Shapes and vocabularies

Resource shapes in OSLC Core 3.0 are upwards compatible with those defined in OSLC Core 2.0. Core 3.0 supports some additional optional constraints that include: 1. On a resource shape, `oslc:hidden` may be used to indicate the type should be hidden. Some OSLC 2.0 servers may have implemented this prior to its inclusion in OSLC Core 3.0. 2. On a property, `oslc:queryable` may be used to indicate whether a property is queryable and can be specified in an `oslc.where` query expression.

OSLC Core 3.0 provides separately fetchable files for resource shapes defined in the specification. These may be fetched as RDF (in RDF/XML or Turtle formats) or as rendered HTML using standard HTTP content negotiation. The OSLC Core 3.0 vocabulary and constraints multi-part specification documents are generated from these RDF ResourceShape constraint representations.

## 11. Authentication

OSLC Core 2.0 supports the use of HTTP Basic Authentication, OAuth (1.0), or both. OSLC Core 3.0 recommends that servers support OAuth 2.0 and OpenIdConnect, and only support HTTP Basic Authentication via SSL. Existing clients that support none of these may be unable to use an OSLC 3.0 server.

## 12. CORS protocol

OSLC Core 2.0 did not address the issue of new security constraints being imposed by some web browsers to limit the loading of resources with cross-origin. See [Cross-origin resource sharing](#). An example is where a dialog served by one application embeds content from another application, such as in an iFrame. OSLC Core 3.0 recommends that servers support the CORS protocol to address these emerging security requirements.



## Appendix A. Acknowledgements

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

### **Project Governing Board:**

James Amsden, IBM (co-chair)  
Andrii Berezovskyi, KTH (co-chair)  
Axel Reichwein, Koneksys

### **Technical Steering Committee:**

James Amsden, IBM  
Andrii Berezovskyi, KTH  
Axel Reichwein, Koneksys

### **Additional Participants:**

David Honey